

Proiect - Baze de Date

Gestiunea unui lanț hotelier

Ciuperceanu Vlad-Mihai

Grupa 151

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.....	4
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.....	5
3. Descrierea entitătilor, incluzând precizarea cheii primare.....	6
4. Descrierea relațiilor, incluzând precizarea cardinalitatii acestora.....	7
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.....	9
6. Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5....	14
7. Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectata la punctul 6. Diagrama conceptuală obținuta trebuie să contină minimum 6 tabele (fara considerarea subentitătilor), dintre care cel putin un tabel asociativ.....	15
8. Enumerarea schemelor relationale corespunzatoare diagramei conceptuale proiectata la punctul 7.....	16
9. Realizarea normalizării pana la forma normală 3 (FN1-FN3).....	16
10. Crearea unei sechente ce va fi utilizata in inserarea inregistrarilor in tabele (punctul 11).....	18
11. Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea (minimum 5 inregistrari in fiecare tabel neasociativ; minimum 10 inregistrari in tabelele asociative).....	19
12. Formulati in limbaj natural si implementati 5 cereri SQL complexe ce vor utiliza, in ansamblul lor, urmatoarele elemente:	
• subcereri sincronizate in care intervin cel putin 3 tabele	
• subcereri nesincronizate in clauza FROM	
• grupari de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)	
• ordonari si utilizarea functiilor NVL si DECODE (in cadrul aceleiasi cereri)	
• utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE	
• utilizarea a cel putin 1 bloc de cerere (clauza WITH).....	57
13. Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri.....	64

14. Crearea unei vizualizari complexe. Dati un exemplu de operație LMD permisa pe vizualizarea respectiva si un exemplu de operatie LMD nepermisa.....	69
15. Formulati in limbaj natural si implementati in SQL: o cerere ce utilizeaza operatia outer-join pe minimum 4 tabele, o cerere ce utilizeaza operatia division si o cerere care implementeaza analiza top-n.....	69
16. Optimizarea unei cereri, aplicand regulile de optimizare ce deriva din proprietatile operatorilor algebrei relationale. Cererea va fi exprimata prin expresie algebraica, arbore algebraic si limbaj (SQL), atat anterior cat si ulterior optimizarii.....	75
17.	
a. Realizarea normalizarii BCNF, FN4, FN5	
b. Aplicarea denormalizarii, justificand necesitatea acesteia.....	79

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Modelul de date va gestiona informații legate de modalitatea de funcționare a unui lanț hotelier, reținând informații atât despre structura internă a unui hotel, cât și despre parteneriatele acestuia și rezervările făcute la hotelul respectiv.

Fiecare hotel se află la o locație și are mai multe anexe proprii, ce propun servicii destinate clienților. Pentru întreținerea și buna funcționare a acestora, hotelurile pot fi aprovizionate de mai mulți parteneri ce pun la dispoziție anumite tipuri de produse contra cost.

De asemenea, fiecare hotel oferă anumite pachete de cumpărare, ce prezintă beneficii pentru clienți, legate de serviciile angajaților și de folosirea anexelor. Acestea joacă un rol important și în crearea unei rezervări, întrucât clienții trebuie să aleagă un pachet împreună cu o cameră pentru a-și finaliza cazarea.

În cadrul unui hotel lucrează mai mulți angajați, având diverse joburi. Există și o ierarhie printre aceștia, fiind unii angajați care au alții angajați în subordine.

Toate hotelurile au mai multe camere, ce pot fi de mai multe tipuri, în funcție de nevoile și dorințele clienților. În același timp, fiecare tip de cameră oferă anumite facilități, după care se pot ghida clienții în decizia pe care o vor lua.

Rezervările sunt făcute de către clienți, alegându-și camerele și beneficiile dorite în funcție de pachetele puse la dispoziție de către hotelul la care se dorește cazarea. Este nevoie, pentru finalizarea unei rezervări, să existe o cameră și un pachet pentru care a optat clientul, precum și precizarea datei de sosire.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Modelul de date prezintă anumite restricții (constrângerii) de funcționare – acestea se implementează la nivel de cardinalitate:

- Fiecare hotel se află la o adresă.
- Este posibil ca la o adresă să nu se regăsească niciun hotel.
- Un partener poate aproviziona mai multe hoteluri, dar nu este neapărat să aprovizioneze vreunul.
- Fiecare hotel poate fi aprovizionat de mai mulți parteneri, dar este posibil ca un hotel să nu fie aprovizionat de niciun partener.
- O anexă poate apartine de un singur hotel.
- Pot exista hoteluri care să nu aibă anexe.
- În cadrul unui hotel lucrează cel puțin un angajat.
- Fiecare angajat poate apartine de un singur hotel.
- Un angajat poate conduce mai mulți alți angajați, dar poate să nu conducă pe nimeni.
- Este posibil ca un angajat să nu fie condus de niciun angajat.
- Un hotel oferă cel puțin un pachet.
- Un pachet poate fi oferit de un singur hotel în forma sa completă.
- Fiecare cameră aparține unui singur hotel.
- Un hotel are cel puțin o cameră.
- Pentru un tip de cameră pot exista mai multe camere, dar nu este neapărat să existe vreuna.
- O cameră este de un singur tip.
- Un tip de cameră oferă mai multe facilități, dar nu este neapărat să aibă o facilitate.
- O facilitate poate fi oferită de mai multe tipuri de cameră, fiind obligatoriu să existe măcar un tip de cameră care să o ofere.
- Un client poate să își rezerve mai multe camere folosind mai multe pachete, la mai multe date, fiind obligatoriu ca pentru o rezervare să existe un client, o cameră, un pachet și o dată a sosirii.

3. Descrierea entitătilor, incluzând precizarea cheii primare.

HOTEL = entitate ce identifică un hotel. Cheia primară a entității este *cod_hotel*.

LOCATIE = entitate care identifică locația la care se află un hotel. Cheia primară a entității este *cod_locatie*.

PARTENER = persoană juridică ce aprovizionează hotelul cu diverse produse. Cheia primară a entității este *cod_partener*.

ANGAJAT = persoană fizică, angajată la un hotel, care este responsabilă cu anumite activități destinate hotelului. Cheia primară a acestei entități este *cod_angajat*.

CAMERA = entitate din cadrul unui hotel, identificând camerele în care sunt cazați clienții. Aceasta prezintă anumite facilități, depinzând de tipul acesteia. Cheia primară a acestei entități este *cod_camera*.

PACHET = entitate care identifică pachetele de rezervare oferite de hotel. Cheia primară a acestei entități este *cod_pachet*.

TIP_CAMERA = entitate care identifică tipul unei camere. Cheia primară a acestei entități este *cod_tip*.

FACILITATE = entitate care identifică facilitatea pe care o poate oferi un tip de cameră. Cheia primară a acestei entități este *cod_facilitate*.

ANEXA = entitate din cadrul unui hotel, ce identifică anexele pe care le are un hotel. Cheia primară a acestei entități este compusă din *cod_anexa*.

CLIENT = persoană fizică, ce dorește să facă rezervări ale unor camere, cu un anumit pachet, la un hotel. Cheia primară a acestei entități este *cod_client*.

4. Descrierea relațiilor, incluzând precizarea cardinalitatii acestora.

PARTENER_aprovizioneaza_HOTEL = relație dintre entitățile PARTENER și HOTEL, reprezentând aprovizionarea făcută de către un partener pentru un hotel. Relație de cardinalitate minimă 0:0 (un partener nu trebuie neapărat să aprovizioneze un hotel, iar un hotel nu trebuie neapărat să fie aprovizionat de un partener) și cardinalitate maximă m:n (un partener poate aprovizia mai multe hoteluri, iar un hotel poate fi aprovizionat de mai mulți parteneri).

HOTEL_se_afla_la_LOCATIE = relație dintre entitățile HOTEL și LOCATIE, reprezentând locația la care se află un hotel. Relație de cardinalitate minimă 0:1 (un hotel trebuie să aibă o locație, dar nu trebuie neapărat ca la o locație să se afle un hotel) și cardinalitate maximă 1:1 (un hotel se poate afla la o singură locație, iar la o anumită locație putem găsi cel mult un hotel).

ANGAJAT_lucreaza_in_HOTEL = relație dintre entitățile ANGAJAT și HOTEL, reprezentând faptul că angajații lucrează în cadrul unui hotel. Relație de cardinalitate minimă 1:1 (un angajat trebuie să lucreze la un hotel, iar un hotel trebuie să aibă măcar un angajat) și cardinalitate maximă n:1 (un angajat trebuie să lucreze la exact un hotel, dar un hotel poate avea mai mulți angajați).

ANGAJAT_conduce_ANGAJAT = relație dintre entitățile ANGAJAT și ANGAJAT, reprezentând faptul că unii angajați pot conduce alții angajați. Relație de cardinalitate minimă 0:0 (un angajat nu trebuie să conducă alții angajați, iar un angajat nu trebuie să fie condus de un alt angajat) și cardinalitate maximă n:1 (un angajat poate fi condus de exact un angajat, dar un angajat poate avea mai mulți angajați în subordine).

HOTEL_detine_CAMERA = relație dintre entitățile HOTEL și CAMERA, reprezentând camerele deținute de un hotel. Relație de cardinalitate minimă 1:1 (un hotel trebuie să aibă o cameră, iar o cameră trebuie să aparțină unui hotel) și cardinalitate maximă 1:n (un hotel poate deține mai multe camere, dar o cameră nu poate să aparțină mai multor hoteluri).

HOTEL_ofera_PACHET = relație dintre entitățile HOTEL și PACHET, reprezentând pachetele de cumpărare oferite de un hotel. Relație de cardinalitate minimă 1:1 (un hotel trebuie să ofere cel puțin un pachet, iar un pachet trebuie să aparțină unui hotel) și cardinalitate maximă 1:n (un hotel poate oferi mai multe pachete, dar un pachet poate fi oferit doar de un singur hotel).

CAMERA_are_TIP_CAMERA = relație dintre entitățile CAMERA și TIP_CAMERA, reprezentând tipurile de cameră posibile. Relație de cardinalitate minimă 0:1 (o cameră trebuie să fie de un anumit tip, dar un tip de cameră nu trebuie neapărat să aibă o cameră de tipul respectiv) și cardinalitate maximă n:1 (o cameră nu se poate încadra la mai multe tipuri de cameră, dar un tip de cameră poate avea mai multe camere de tipul respectiv).

TIP_CAMERA_ofera_FACILITATE = relație dintre entitățile TIP_CAMERA și FACILITATI, reprezentând facilitățile pe care le oferă un tip de cameră. Relație de cardinalitate minimă 1:0 (un tip de cameră nu trebuie neapărat să aibă o facilitate, dar o facilitate trebuie să aparțină cel puțin unei camere) și cardinalitate maximă m:n (un tip de cameră poate oferi mai multe facilități, iar un o facilitate poate fi oferită de mai multe tipuri de cameră).

HOTEL_are_ANEXA = relație dintre entitățile HOTEL și ANEXA, reprezentând anexele pe care le are un hotel. Relație de cardinalitate minimă 1:0 (un hotel nu trebuie neapărat să aibă o anexă, dar o anexă trebuie să aparțină unui hotel) și cardinalitate maximă 1:n (un hotel poate avea mai multe anexe, dar o anexă trebuie să aparțină unui singur hotel).

CLIENT_rezerva_CAMERA_cu_PACHET = relație de tip 3 ce leagă entitățile CLIENT, CAMERA și PACHET, reflectând clientul care a rezervat o cameră, cu un pachet ales dintre cele oferite de hotel. Denumirea acestei relații este rezerva.

5. Descrierea atributelor, incluzand tipul de date si eventualele constrangeri, valori implicate, valori posibile ale atributelor.

Entitatea **HOTEL** are ca atribute:

- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului NUMBER(2)
- cod_locatie = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al locației hotelului NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul LOCATIE.
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele hotelului VARCHAR2(25) și care nu poate fi NULL
- data_infiintare = variabilă de tip dată calendaristică, care reprezintă data înființării hotelului și care are valoarea default egală cu SYSDATE
- nr_camere = variabilă de tip întreg, de lungime maximă 2, care reprezintă numărul de camere cu care a fost proiectat hotelul NUMBER(2)
- stele = variabilă de tip întreg, de lungime maximă 1, care reprezintă numărul de stele ale hotelului NUMBER(1)

Entitatea **LOCATIE** are ca atribute:

- cod_locatie = variabilă de tip intreg, de lungime maximă 2, care reprezintă codul unic al locației NUMBER(2)
- strada = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele străzii din adresa locației VARCHAR2(25) și care nu poate fi NULL
- numar = variabilă de tip intreg, de lungime maximă 3, care reprezintă numărul din adresa locației NUMBER(3) și care nu poate fi NULL
- oras = variabilă de tip caracter, de lungime maximă 10, care reprezintă numele orașului din adresa locației VARCHAR2(10) și care nu poate fi NULL
- tara = variabilă de tip caracter, de lungime maximă 10, care reprezintă numele țării din adresa locației VARCHAR2(10) și care nu poate fi NULL

Entitatea **PARTENER** are ca atribute:

- cod_partener = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al partenerului NUMBER(2)
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele partenerului VARCHAR2(25) și care nu poate avea valoarea NULL
- tip_produse = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele tipului de produse pe care le oferă partenerul VARCHAR2(25) (ex: Curatenie)

Entitatea **ANGAJAT** are ca atribute:

- cod_angajat = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al angajatului NUMBER(4)
- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului la care lucrează angajatul NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul HOTEL.
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului VARCHAR2(25) și care nu poate avea valoarea NULL
- prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului VARCHAR2(25)
- data_nasterii = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv și care nu poate fi NULL
- job = variabilă de tip caracter, de lungime maximă 25, care reprezintă jobul angajatului VARCHAR2(25) și care nu poate fi NULL
- salariu = variabilă de tip întreg, de lungime maximă 5, care reprezintă salariul angajatului NUMBER(5) și care nu poate fi NULL
- cod_manager = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al managerului pe care îl are angajatul NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul ANGAJAT(cod_angajat).

Entitatea **CAMERA** are ca atribute:

- cod_camera = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al camerei NUMBER(4)
- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul HOTEL
- cod_tip = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al tipului camerei NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul TIP_CAMERA
- numar = variabilă de tip întreg, de lungime maximă 2, care reprezintă numărul camerei NUMBER(2) și care nu poate fi NULL
- etaj = variabilă de tip întreg, de lungime maximă 1, care reprezintă etajul camerei NUMBER(1) și care nu poate fi NULL

Entitatea **PACHET** are ca atribute:

- cod_pachet = variabilă de tip întreg, de lungime maximă 3, care reprezintă codul unic al pachetului NUMBER(3)
- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului care pune la dispoziție pachetul NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul HOTEL.
- beneficiu = variabilă de tip caracter, de lungime maximă 50, care reprezintă beneficiul adus de pachet VARCHAR2(50) și care nu poate fi NULL (ex: All Inclusive)
- pret = variabilă de tip întreg, de lungime maximă 5, care reprezintă prețul pachetului NUMBER(5) și care are valoarea default egală cu 100

Entitatea **TIP_CAMERA** are ca atribute:

- cod_tip = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al tipului de camera NUMBER(2)
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele tipului de camera VARCHAR2(25) și care nu poate avea valoarea NULL
- capacitate = variabilă de tip întreg, de lungime maximă 1, care reprezintă capacitatea tipului de camera NUMBER(1) și care nu poate fi NULL (ex: dacă valoarea este 3, înseamnă că acel tip de cameră este o cameră de 3 persoane)
- suprafata = variabilă de tip întreg, de lungime maximă 3, care reprezintă suprafața în metrii pătrați pe care o are tipul de camera NUMBER(3) și care nu poate fi NULL
- pret = variabilă de tip întreg, de lungime maximă 4, care reprezintă prețul tipului de camera NUMBER(4) și care nu poate fi NULL

Entitatea **FACILITATE** are ca atribute:

- cod_facilitate = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al facilității NUMBER(2)
- facilitate = variabilă de tip caracter, de lungime maxima 25, care reprezintă denumirea facilității VARCHAR2(25) și care nu poate fi NULL
- pret = variabilă de tip întreg, de lungime maximă 3, care reprezintă prețul facilității care este inclus în prețul camerei NUMBER(3)

Entitatea **ANEXA** are ca atribute:

- cod_anexa = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al anexei NUMBER(2)
- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului la care se află anexa NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul HOTEL.

- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele anexei VARCHAR2(25) și care nu poate avea valoarea NULL
- capacitate = variabilă de tip întreg, de lungime maximă 2, care reprezintă capacitatea anexei NUMBER(2)

Entitatea **CLIENT** are ca atribute:

- cod_client = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al clientului NUMBER(5)
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului VARCHAR2(25) și care nu poate fi NULL
- prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului VARCHAR2(25)
- CNP = variabilă de tip caracter, de lungime exact 13, care reprezintă codul numeric personal al clientului CHAR(13) și care nu poate fi NULL

Relația **PARTENER_aprovizioneaza_HOTEL** are ca atribute:

- cod_partener = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al partenerului NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul PARTENER.
- cod_hotel = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al hotelului NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul HOTEL.
- data_aprovizionare = variabilă de tip dată calendaristică, reprezentând data aprovizionării (care va fi unică pentru o anumită aprovizionare) și care are valoarea default egală cu SYSDATE
- cost = variabilă de tip întreg, de lungime maximă 5, care reprezintă costul aprovizionării NUMBER(5)

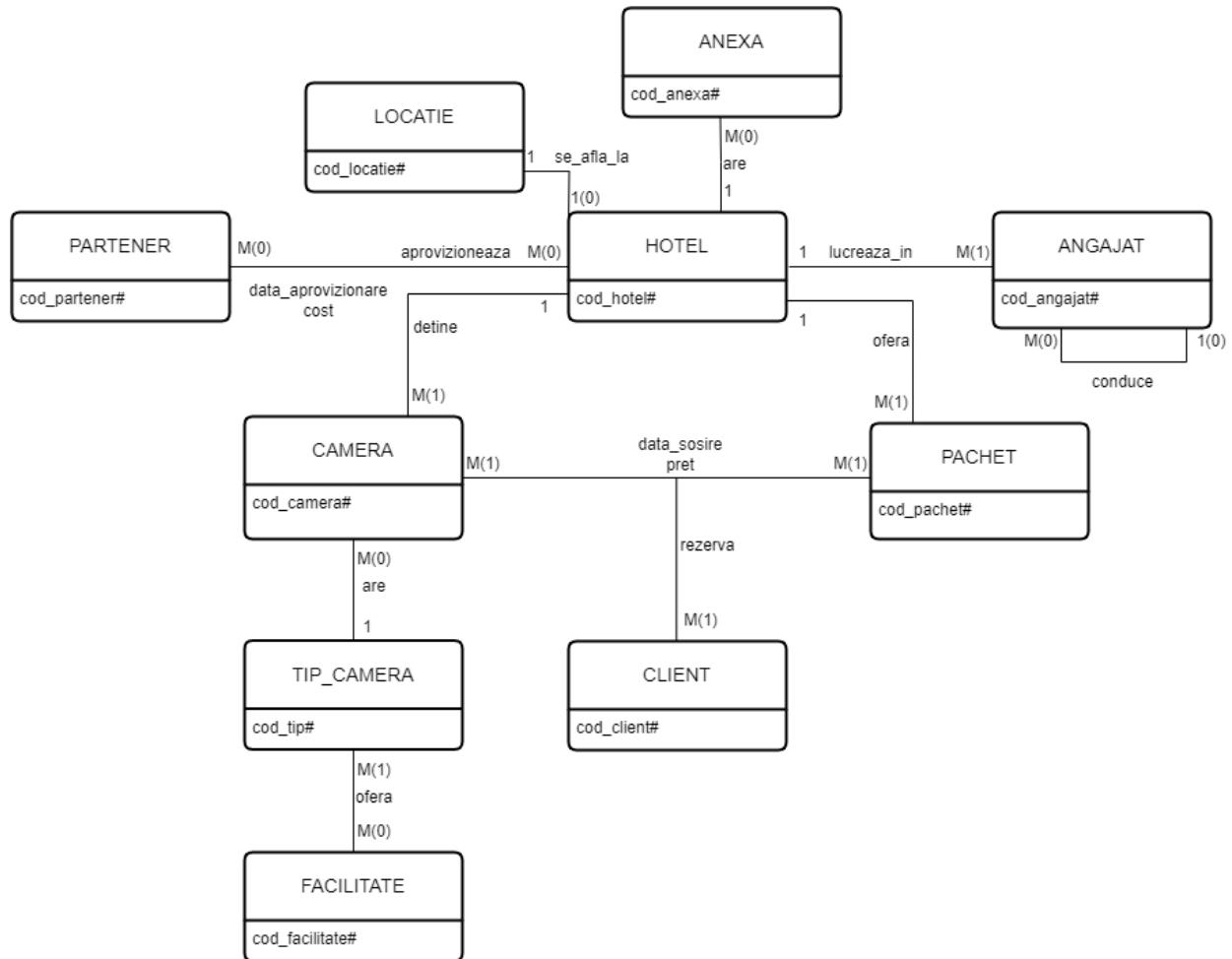
Relatia **TIP_CAMERA_ofera_FACILITATE** are ca atribute:

- cod_tip = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al tipului de camera NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul TIP_CAMERA
- cod_facilitate = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic al facilitatii NUMBER(2). Atributul trebuie să corespundă valorii cheii primare din tabelul FACILITATE

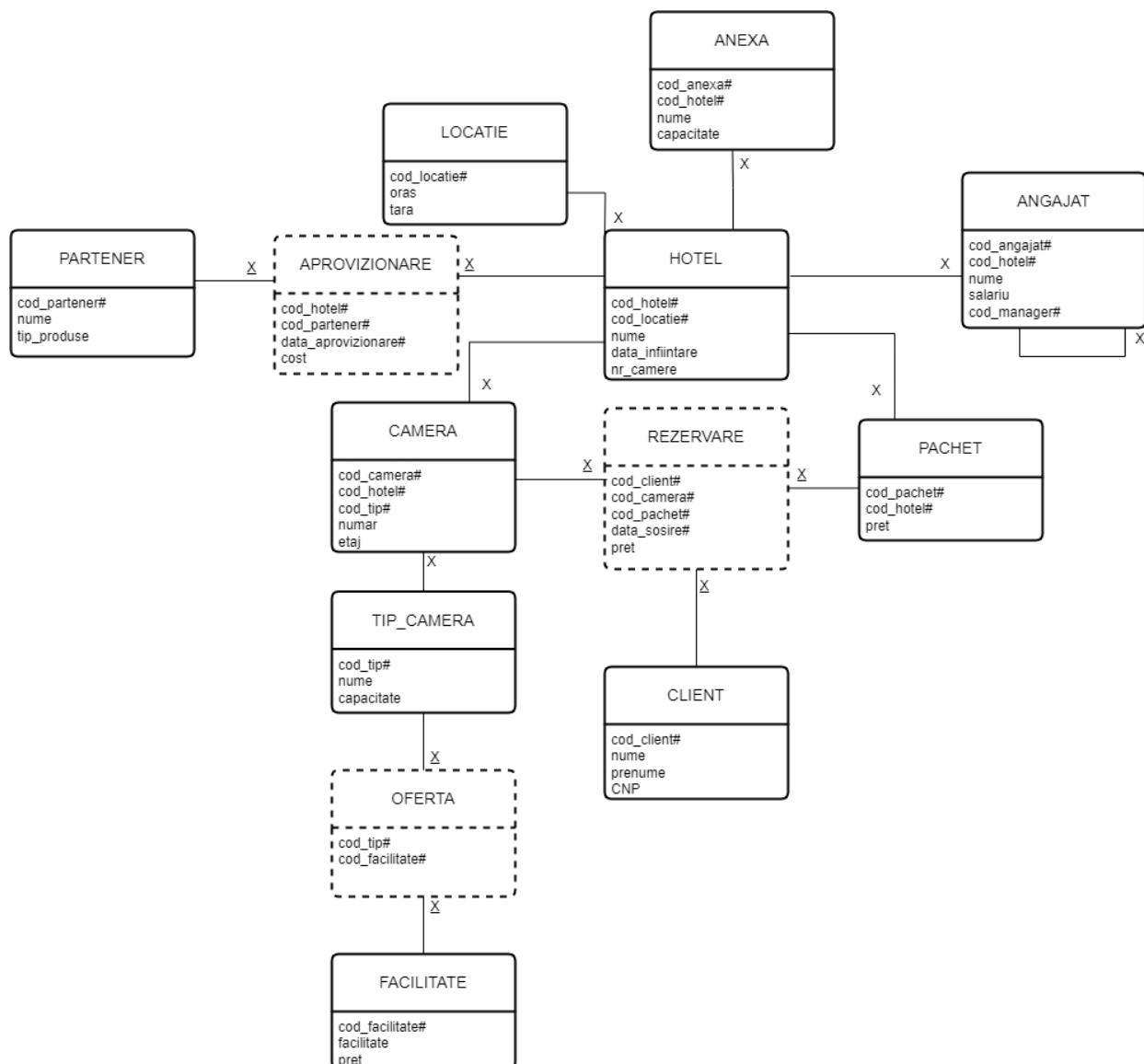
Relatia **CLIENT_rezerva_CAMERA_cu_PACHET** are ca atribute:

- cod_client = variabilă de tip întreg, de lungime maxima 5, care reprezintă codul unic al clientului NUMBER(5). Atributul trebuie să corespundă valorii cheii primare din tabelul CLIENT
- cod_camera = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al camerei NUMBER(4). Atributul trebuie să corespundă valorii cheii primare din tabelul CAMERA
- cod_pachet = variabilă de tip întreg, de lungime maximă 3, care reprezintă codul unic al pachetului NUMBER(3). Atributul trebuie să corespundă valorii cheii primare din tabelul PACHET
- data_sosire = variabilă de tip dată calendaristică, reprezentând data sosirii (care este unică pentru o rezervare) și care are valoarea default egală cu SYSDATE
- pret = variabilă de tip întreg, de lungime maximă 5, care reprezintă prețul plătit pentru rezervare NUMBER(5) și care nu poate fi NULL

6. Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectata la punctul 6. Diagrama conceptuala obtinuta trebuie sa contine minimum 6 tabele (fara considerarea subentitatilor), dintre care cel putin un tabel asociativ.



8. Enumerarea schemelor relationale corespunzatoare diagramei conceptuale proiectata la punctul 7.

HOTEL(cod_hotel#, cod_locatie#, nume, data_infiintare, nr_camere, stele)
LOCATIE(cod_locatie#, strada, numar, oras, tara)
ANEXA(cod_anexa#, cod_hotel#, nume, capacitate)
ANGAJAT(cod_angajat#, cod_hotel#, nume, prenume, data_nasterii, job, salariu, cod_manager#)
PARTENER(cod_partener#, nume, tip_produse)
APROVIZIONARE(cod_partener#, cod_hotel#, data_aprovizionare#, cost)
CAMERA(cod_camera#, cod_hotel#, cod_tip#, numar, etaj)
PACHET(cod_pachet#, cod_hotel#, beneficiu, pret)
CLIENT(cod_client#, nume, prenume, CNP)
REZERVARE(cod_client#, cod_camera#, cod_pachet#, data_sosire#, pret)
TIP_CAMERA(cod_tip#, nume, capacitate, suprafata, pret)
OFERTA(cod_tip#, cod_facilitate#)
FACILITATE(cod_facilitate#, facilitate, pret)

9. Realizarea normalizarii pana la forma normala 3 (FN1-FN3).

Diagrama conceptuală descrisă anterior se află deja în forma normală 3, astfel că vom prezenta 3 situații în care ar fi fost nevoie de normalizare, precum și soluțiile prin care putem să aducem relațiile la forma normală dorită.

Exemplu pentru FN1:

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă / atomică.

Observăm că niciun tabel (și nicio relație) din diagrame nu are atrbute cărora să le corespundă valori divizibile, deci diagrama se află în FN1.

Din diagrama E/R, observăm că fiecare hotel are mai multe camere, având o relație de tipul *one-to-many*. Un scenariu în care tabelul HOTEL ar fi făcut ca diagrama să nu fie în FN1 s-ar fi întâmplat dacă am fi reținut în cadrul său un atribut *coduri_camerelor* prin care să păstrăm toate codurile camerelor din hotelul respectiv. Astfel,

tabelul HOTEL ar fi avut un atribut divizibil (anume *coduri_camere*), ceea ce contravine cu FN1.

O soluție este aceea prezentă și în diagrama de mai sus, anume să păstrăm *cod_hotel* drept cheie externă în cadrul tabelului CAMERA, renunțând la atributul *coduri_camere*. Astfel, nu pierdem informația și nici nu mai avem attribute divizibile.

Exemplu pentru FN2:

O relație se află în FN2 dacă:

- 1) se află în FN1 (ceea ce am discutat la exemplul anterior)
- 2) fiecare atribut care nu participă la cheia primară este dependent de întreaga cheie primară

Observăm că singurele tabele cu chei primare compuse sunt cele asociative: APROVIZIONARE, OFERTA, REZERVARE. Restul, având chei primare singulare, se află cu siguranță în FN2.

Tabelul OFERTA nu are attribute care să nu participe la cheia primară compusă. Tabelul APROVIZIONARE mai are ca atribut *cost*, care depinde de întreaga cheie, nu de părți ale sale. De asemenea, tabelul REZERVARE mai are ca atribut *pret*, ce depinde de întreaga cheie.

Un scenariu în care diagrama nu s-ar mai afla în FN2 ar fi fost dacă în tabelul REZERVARE am ține un câmp pentru numele clientului care a realizat rezervarea. Atunci acel câmp ar depinde doar de *cod_client*, nu de toată cheia primară, ce ar duce la redundanță și nu ar mai păstra tabelul în FN2.

O soluție este dată de implementarea din diagramă, în care eliminăm acel atribut și reținem doar cheia primară a clientului ca o componentă a cheii primare compuse din tabelul REZERVARE, soluție ce reține aceeași informație și păstrează FN2.

Exemplu pentru FN3:

O relație se află în FN3 dacă:

- 3) se află în FN2 (ceea ce am discutat la exemplul anterior)
- 4) fiecare atribut care nu participă la o cheie depinde direct de cheia primară

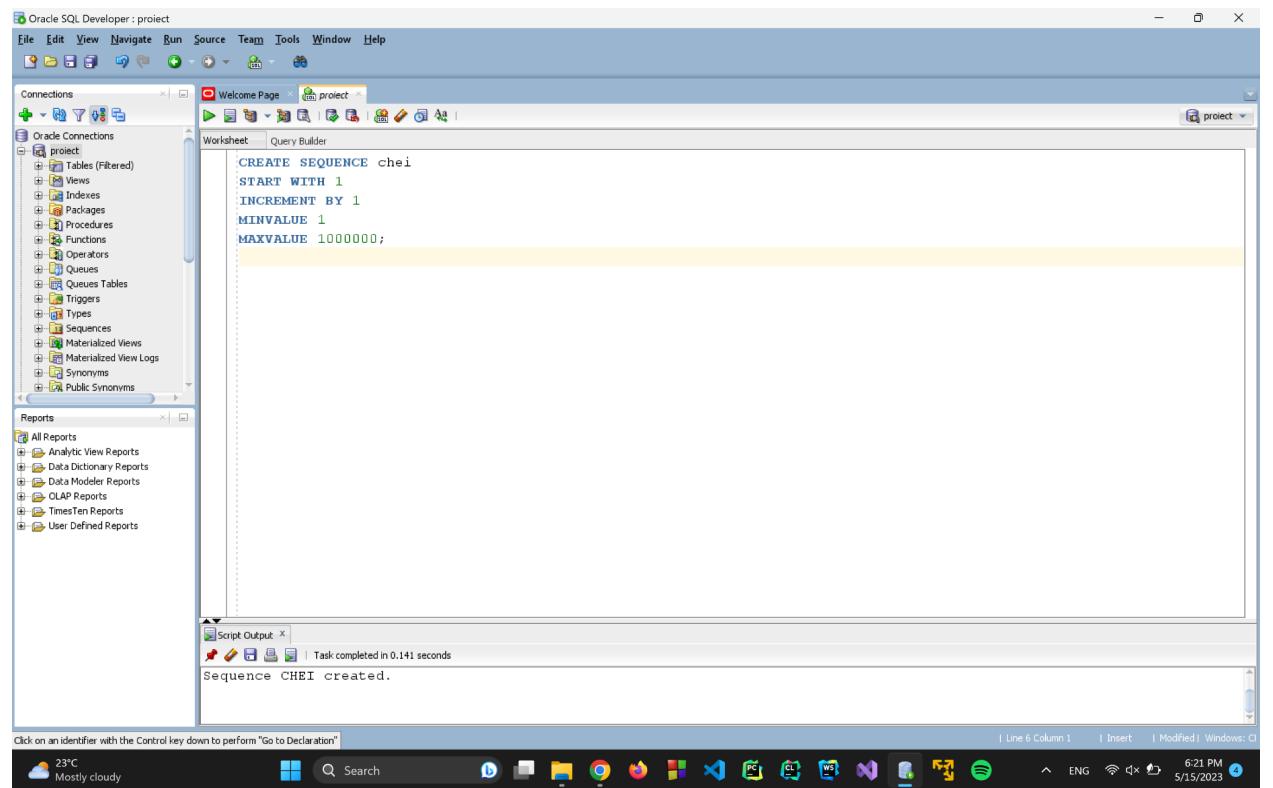
Diagrama prezentată mai sus se află în FN3, deoarece, din modelul de proiectare, nu există dependențe tranzitive.

Un scenariu în care nu s-ar mai păstra forma normală 3 ar fi dacă în tabelul CAMERA am fi păstrat attributele *capacitate*, *suprafata* și *pret*, în loc să le ținem în tabelul TIP_CAMERA. Din felul în care a fost concepută baza de date, aceste attribute țin de tipul de cameră. Astfel, în cadrul tabelului CAMERA, cele 3 attribute menționate mai sus ar depinde tranzitiv de cheia primară prin intermediul atributului *cod_tip*.

Soluția este dată de mutarea attributelor în tabelul TIP_CAMERA, aşa cum am făcut mai sus, menținând relațiile în FN3.

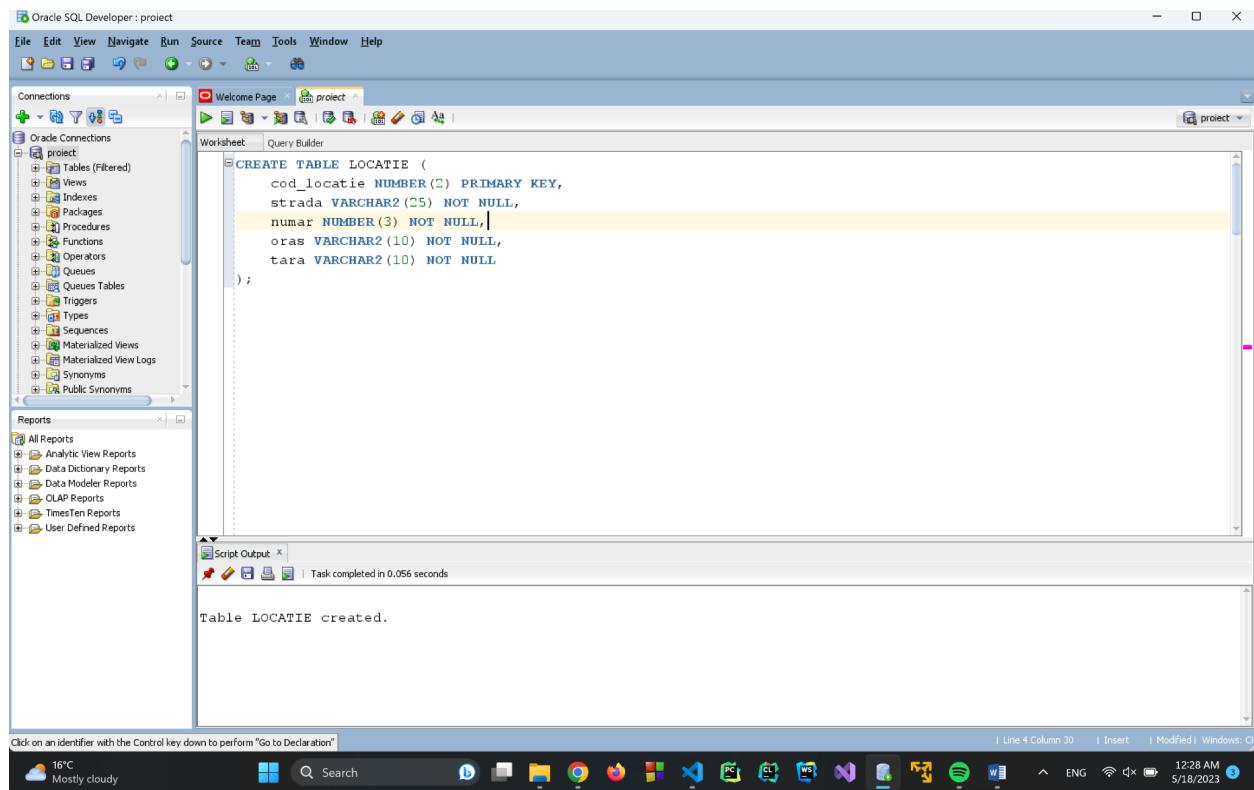
10. Crearea unei sechete ce va fi utilizata in inserarea inregistrarilor in tabele (punctul 11).

```
CREATE SEQUENCE chei  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
MAXVALUE 1000000;
```



11. Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea (minimum 5 inregistrari in fiecare tabel neasociativ; minimum 10 inregistrari in tabelele asociative).

```
CREATE TABLE LOCATIE (
    cod_locatie NUMBER(2) PRIMARY KEY,
    strada VARCHAR2(25) NOT NULL,
    numar NUMBER(3) NOT NULL,
    oras VARCHAR2(10) NOT NULL,
    tara VARCHAR2(10) NOT NULL
);
```



```
CREATE TABLE HOTEL (
    cod_hotel NUMBER(2) PRIMARY KEY,
    cod_locatie NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    data_infiintare DATE DEFAULT SYSDATE,
    nr_camere NUMBER(2),
```

```

stele NUMBER(1),
CONSTRAINT fk_hotel FOREIGN KEY (cod_locatie) REFERENCES
LOCATIE(cod_locatie)
);

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists 'project' under 'Oracle Connections'. The 'Reports' sidebar lists various report types. The central 'Worksheet' tab contains the SQL code for creating the 'HOTEL' table:

```

CREATE TABLE HOTEL (
    cod_hotel NUMBER(2) PRIMARY KEY,
    cod_locatie NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    data_infiintare DATE DEFAULT SYSDATE,
    nr_camere NUMBER(2),
    stele NUMBER(1),
    CONSTRAINT fk_hotel FOREIGN KEY (cod_locatie) REFERENCES LOCATIE(cod_locatie)
);

```

The 'Script Output' window at the bottom shows the results of the execution:

```

Table LOCATIE created.

Table HOTEL created.

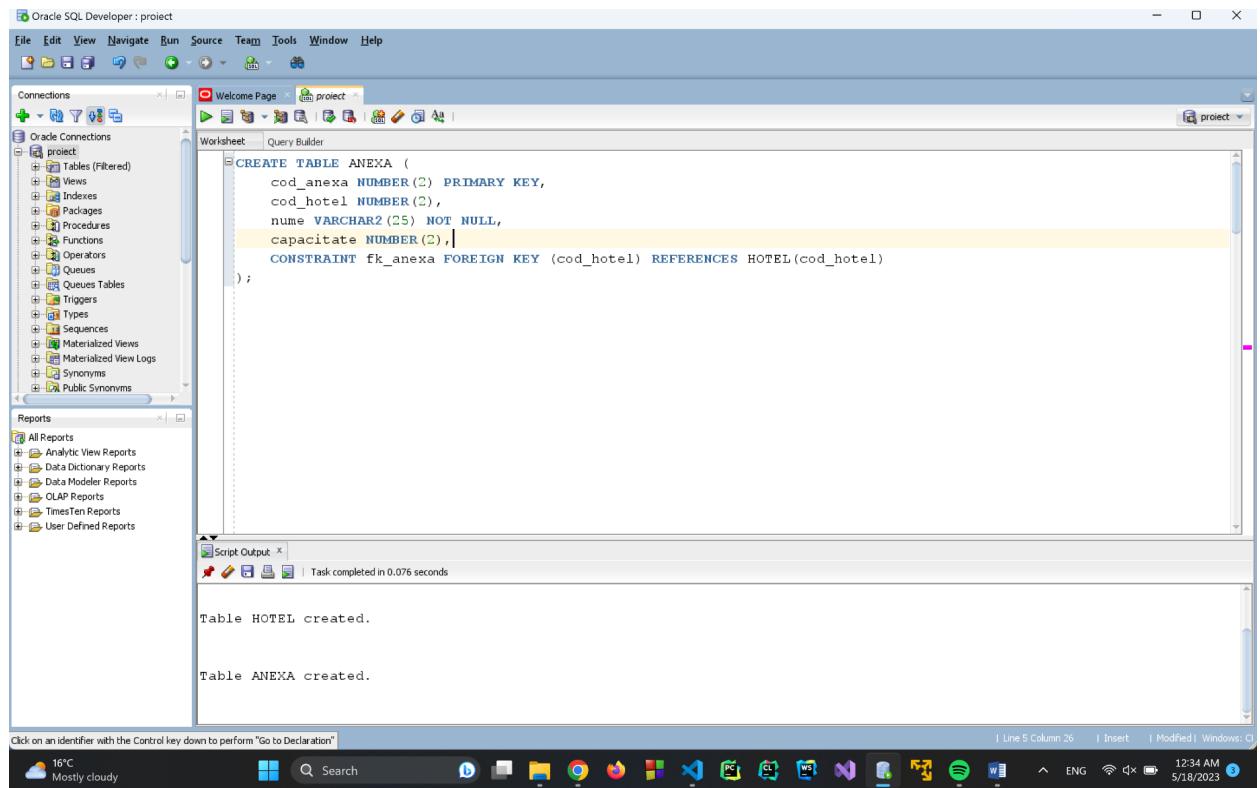
```

The status bar at the bottom right indicates the task completed in 0.188 seconds, the current date and time (5/18/2023 12:32 AM), and system information like weather (16°C, Mostly cloudy).

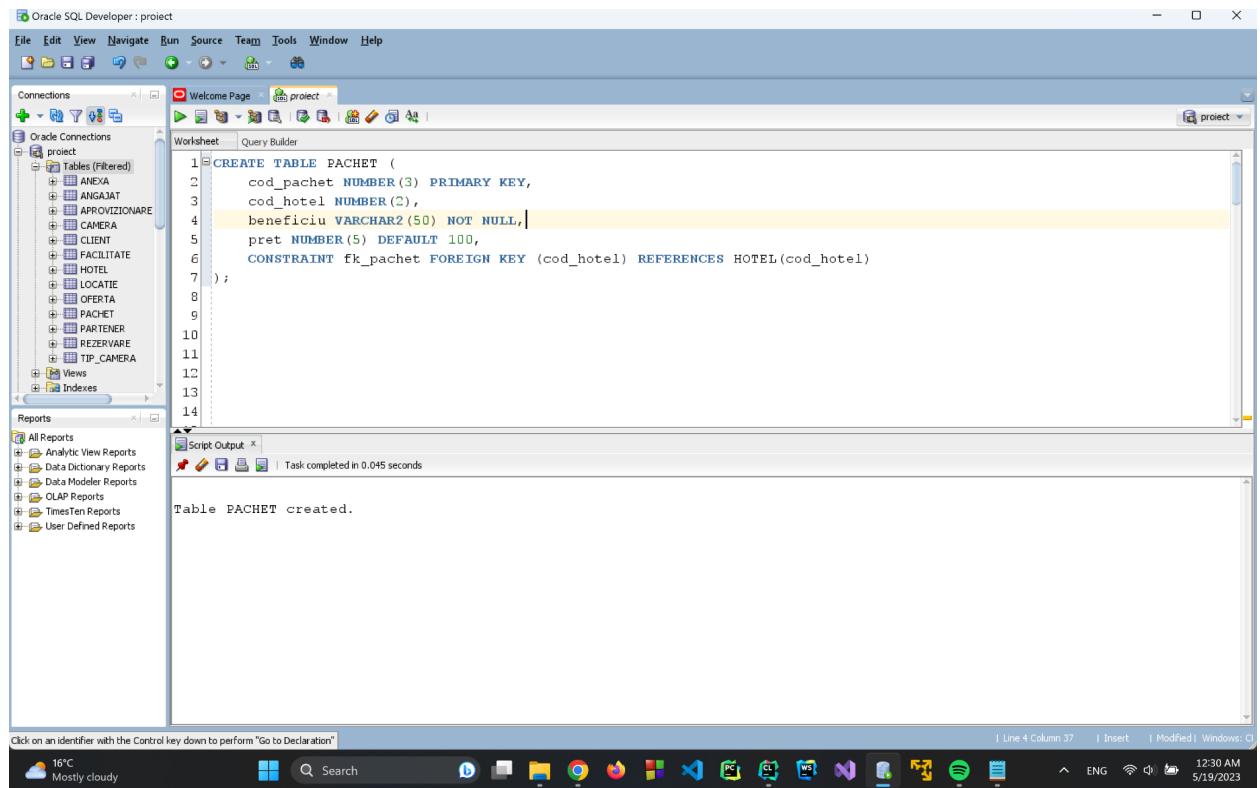
```

CREATE TABLE ANEXA (
    cod_anexa NUMBER(2) PRIMARY KEY,
    cod_hotel NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    capacitate NUMBER(2),
    CONSTRAINT fk_anexa FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel)
);

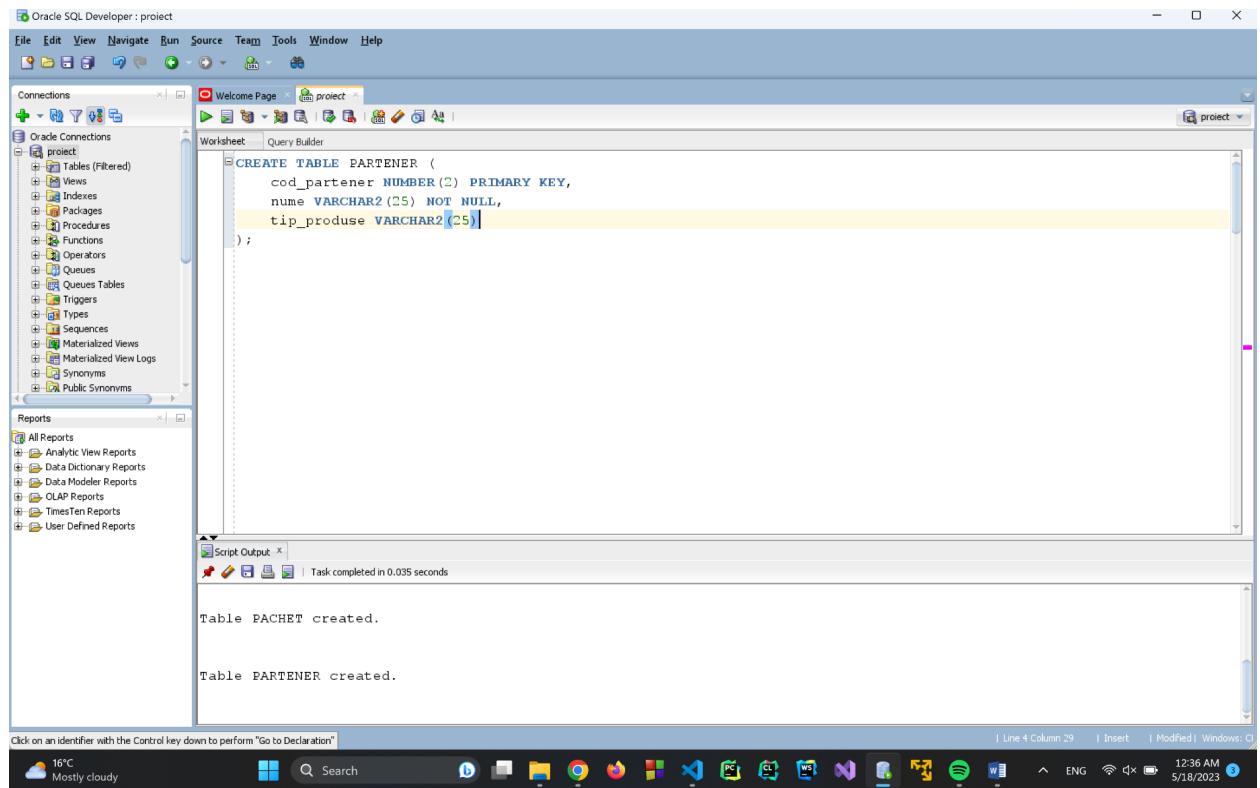
```



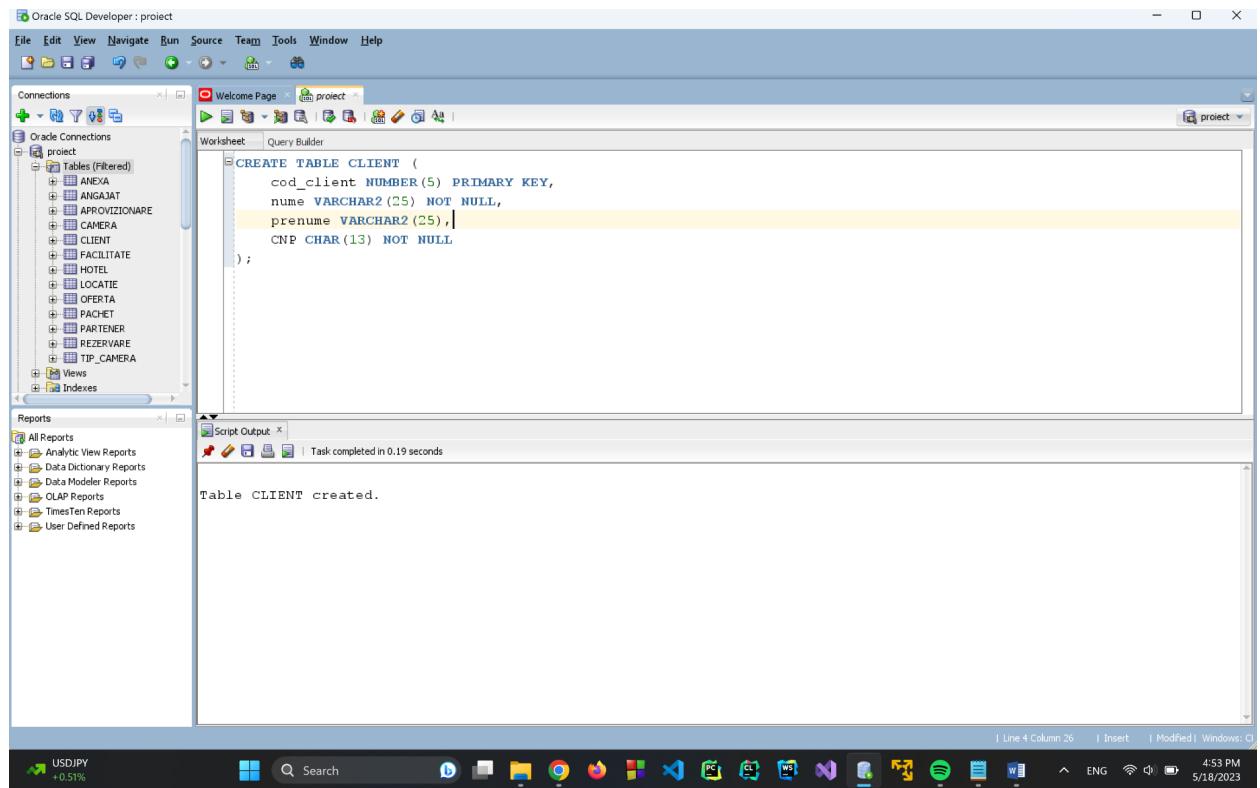
```
CREATE TABLE PACHET (
    cod_pachet NUMBER(3) PRIMARY KEY,
    cod_hotel NUMBER(2),
    beneficiu VARCHAR2(50) NOT NULL,
    pret NUMBER(5) DEFAULT 100,
    CONSTRAINT fk_pachet FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel)
);
```



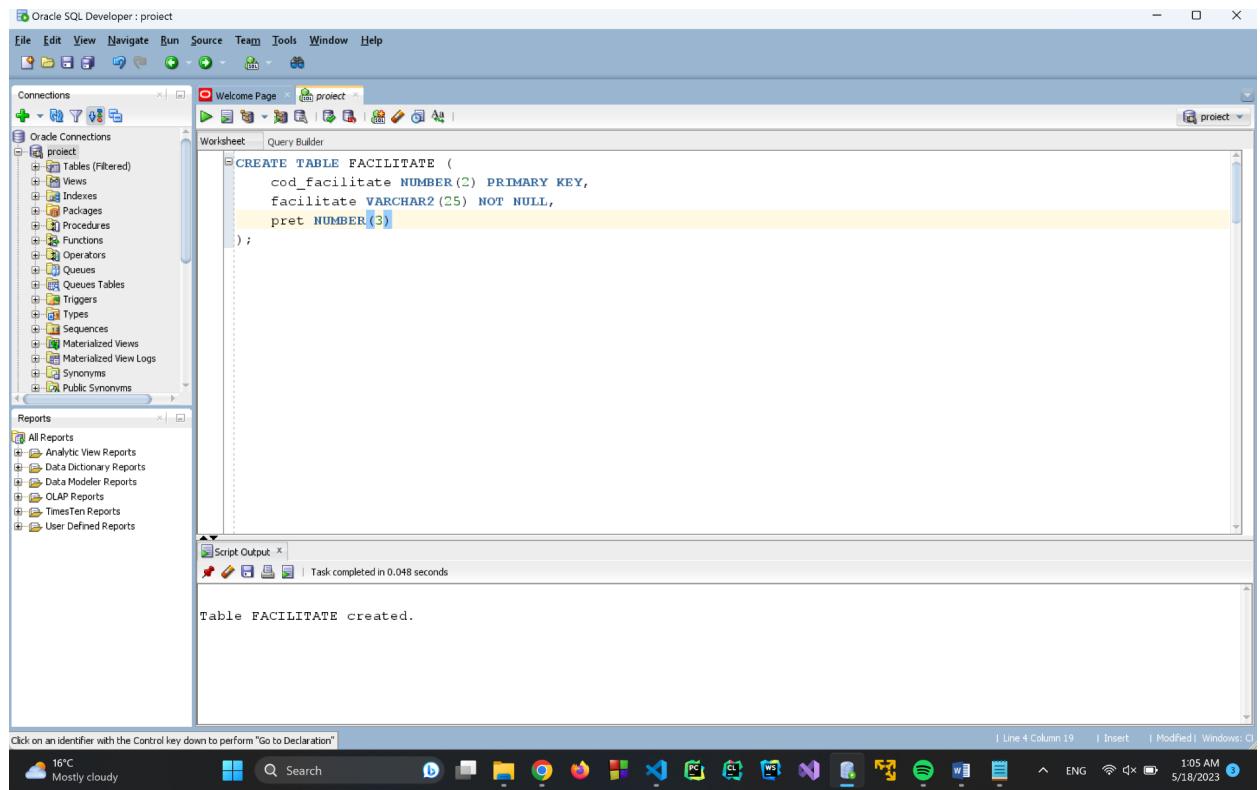
```
CREATE TABLE PARTENER (
    cod_partener NUMBER(2) PRIMARY KEY,
    nume VARCHAR2(25) NOT NULL,
    tip_produse VARCHAR2(25)
);
```



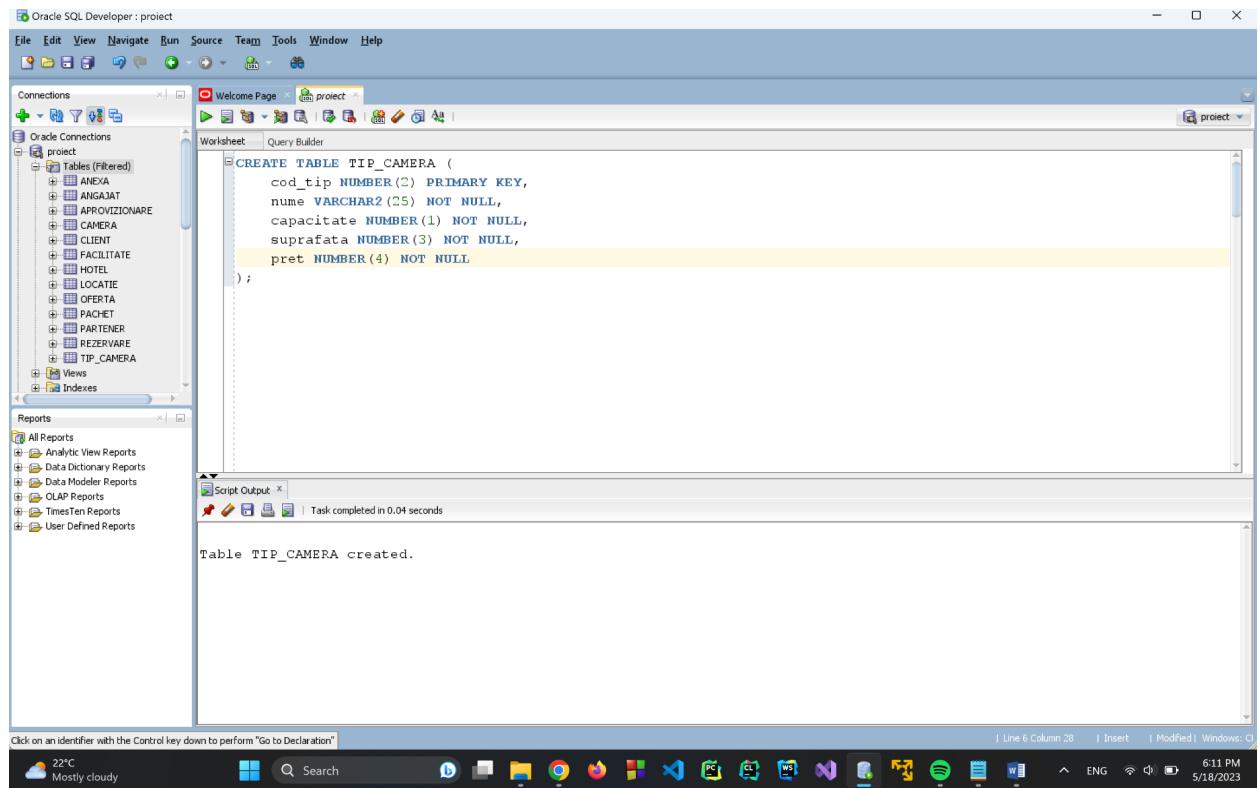
```
CREATE TABLE CLIENT (
    cod_client NUMBER(5) PRIMARY KEY,
    nume VARCHAR2(25) NOT NULL,
    prenume VARCHAR2(25),
    CNP CHAR(13) NOT NULL
);
```



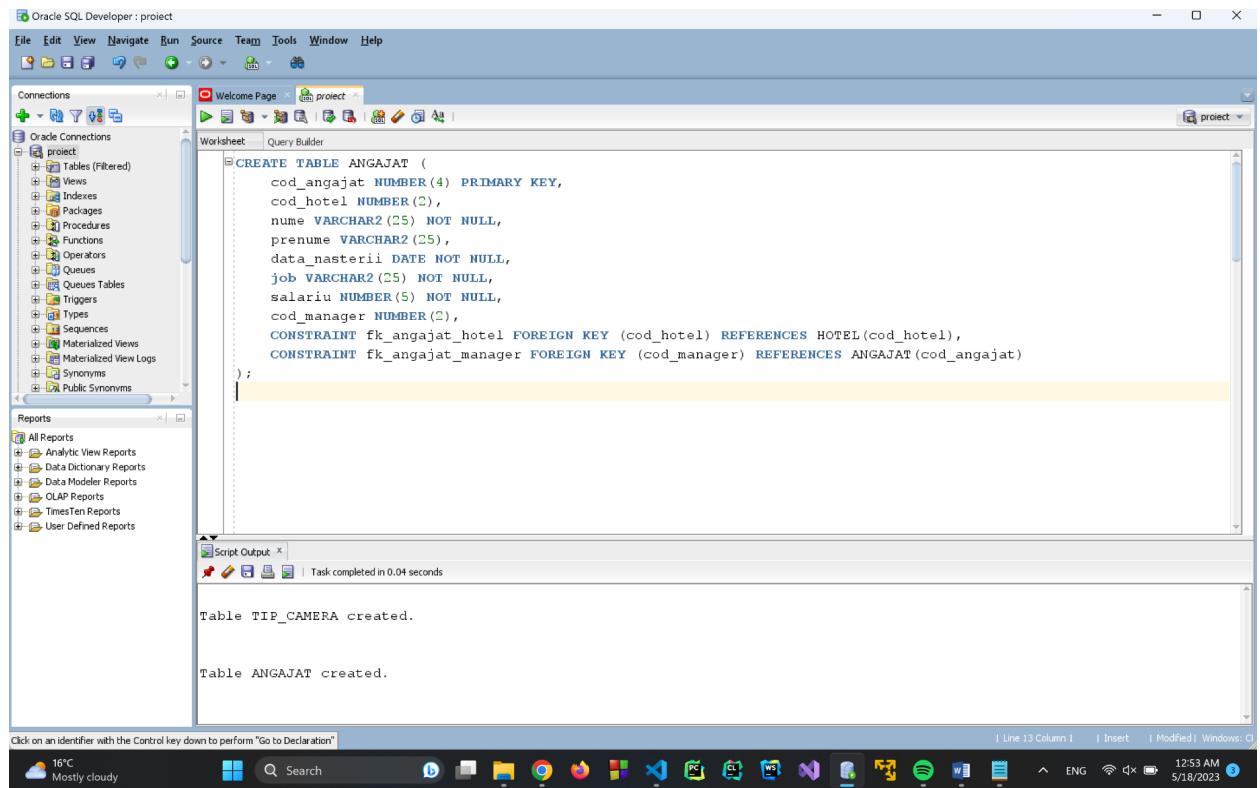
```
CREATE TABLE FACILITATE (
    cod_facilitate NUMBER(2) PRIMARY KEY,
    facilitate VARCHAR2(25) NOT NULL,
    pret NUMBER(3)
);
```



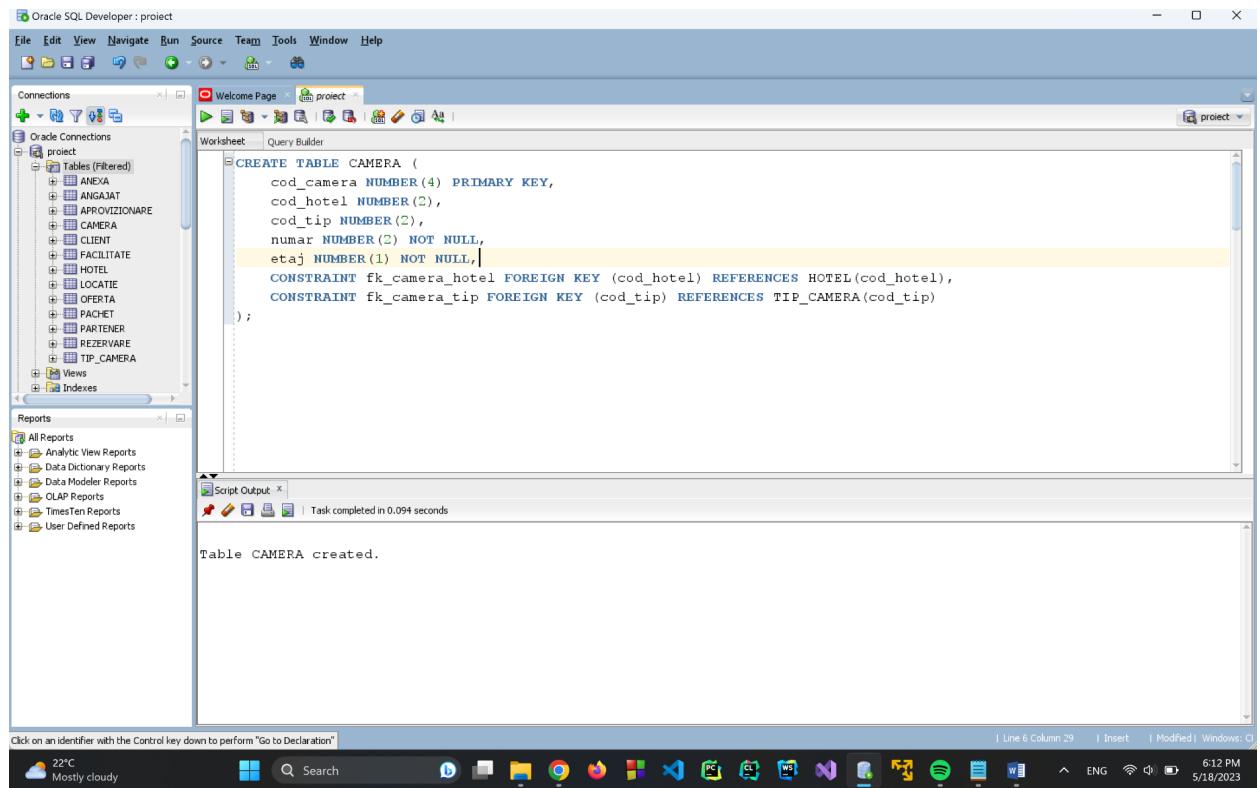
```
CREATE TABLE TIP_CAMERA (
    cod_tip NUMBER(2) PRIMARY KEY,
    nume VARCHAR2(25) NOT NULL,
    capacitate NUMBER(1) NOT NULL,
    suprafata NUMBER(3) NOT NULL,
    pret NUMBER(4) NOT NULL
);
```



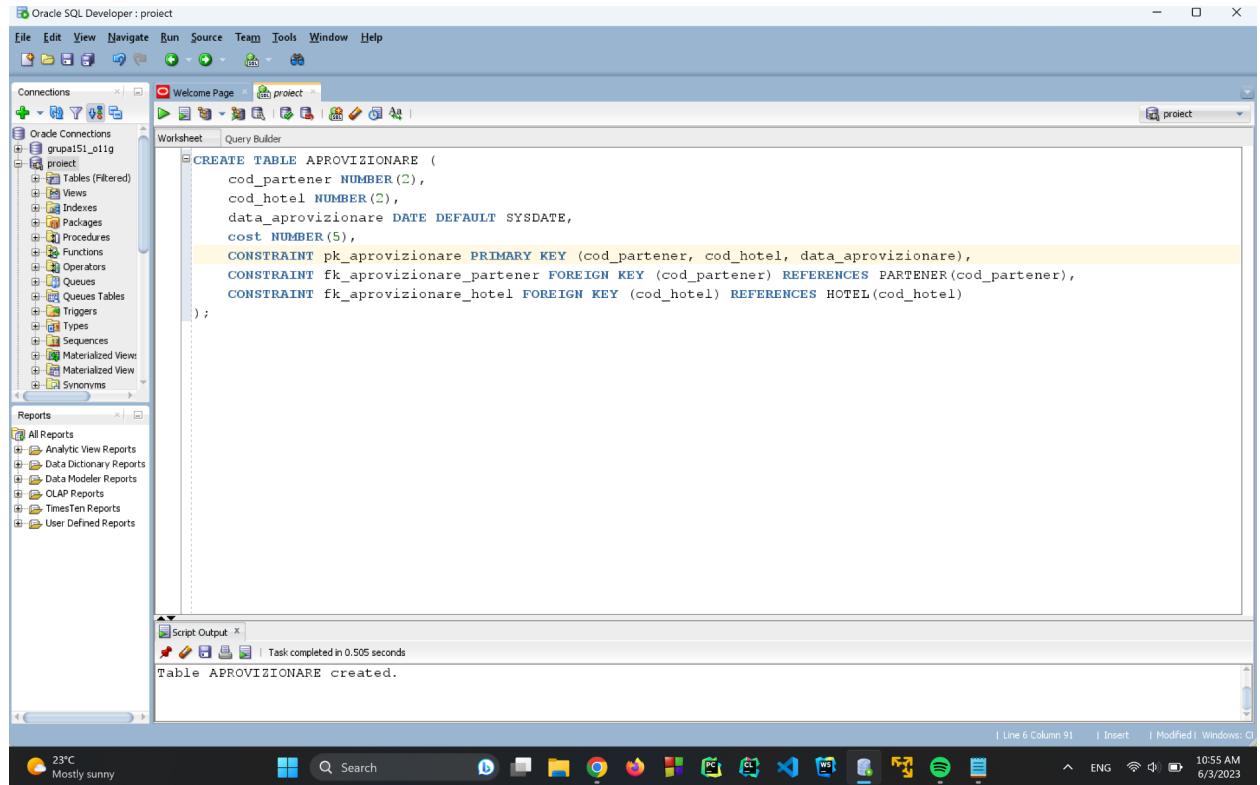
```
CREATE TABLE ANGAJAT (
    cod_angajat NUMBER(4) PRIMARY KEY,
    cod_hotel NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    prenume VARCHAR2(25),
    data_nasterii DATE NOT NULL,
    job VARCHAR2(25) NOT NULL,
    salariu NUMBER(5) NOT NULL,
    cod_manager NUMBER(2),
    CONSTRAINT fk_angajat_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel),
    CONSTRAINT fk_angajat_manager FOREIGN KEY (cod_manager) REFERENCES
ANGAJAT(cod_angajat)
);
```



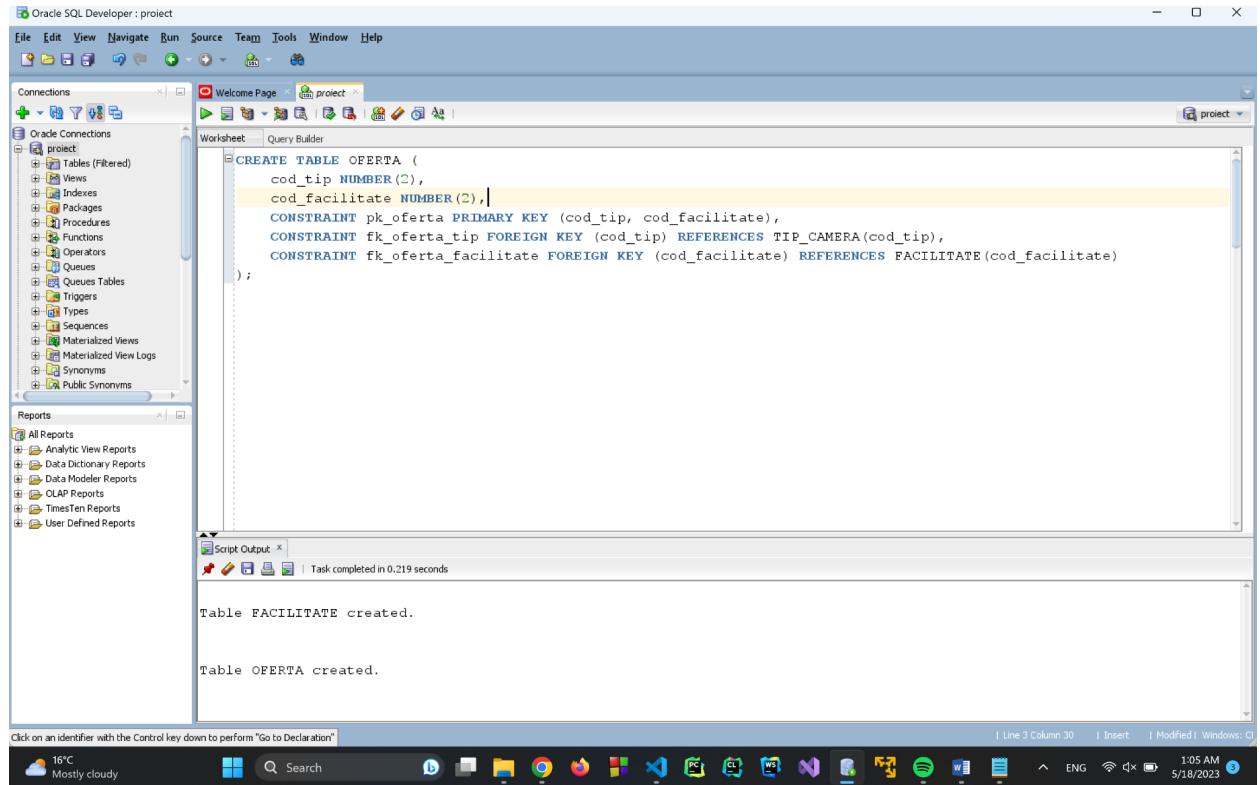
```
CREATE TABLE CAMERA (
    cod_camera NUMBER(4) PRIMARY KEY,
    cod_hotel NUMBER(2),
    cod_tip NUMBER(2),
    numar NUMBER(2) NOT NULL,
    etaj NUMBER(1) NOT NULL,
    CONSTRAINT fk_camera_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel),
    CONSTRAINT fk_camera_tip FOREIGN KEY (cod_tip) REFERENCES
TIP_CAMERA(cod_tip)
);
```



```
CREATE TABLE APROVIZIONARE (
    cod_partener NUMBER(2),
    cod_hotel NUMBER(2),
    data_aprovizionare DATE DEFAULT SYSDATE,
    cost NUMBER(5),
    CONSTRAINT pk_aprovizionare PRIMARY KEY (cod_partener, cod_hotel,
data_aprovizionare),
    CONSTRAINT fk_aprovizionare_partener FOREIGN KEY (cod_partener)
REFERENCES PARTENER(cod_partener),
    CONSTRAINT fk_aprovizionare_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel)
);
```



```
CREATE TABLE OFERTA (
    cod_tip NUMBER(2),
    cod_facilitate NUMBER(2),
    CONSTRAINT pk_oferta PRIMARY KEY (cod_tip, cod_facilitate),
    CONSTRAINT fk_oferta_tip FOREIGN KEY (cod_tip) REFERENCES
TIP_CAMERA(cod_tip),
    CONSTRAINT fk_oferta_facilitate FOREIGN KEY (cod_facilitate) REFERENCES
FACILITATE(cod_facilitate)
);
```



```

CREATE TABLE REZERVARE (
    cod_client NUMBER(5),
    cod_camera NUMBER(4),
    cod_pachet NUMBER(3),
    data_sosire DATE DEFAULT SYSDATE,
    pret NUMBER(5) NOT NULL,
    CONSTRAINT pk_rezervare PRIMARY KEY (cod_client, cod_camera, cod_pachet,
data_sosire),
    CONSTRAINT fk_rezervare_client FOREIGN KEY (cod_client) REFERENCES
CLIENT(cod_client),
    CONSTRAINT fk_rezervare_camera FOREIGN KEY (cod_camera) REFERENCES
CAMERA(cod_camera),
    CONSTRAINT fk_rezervare_pachet FOREIGN KEY (cod_pachet) REFERENCES
PACHET(cod_pachet)
);

```

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, the path 'File Edit View Navigate Run Source Team Tools Window Help' is visible. Below the menu is a toolbar with various icons. On the left, the 'Connections' sidebar shows a project named 'project' containing tables like ANEXA, ANGAJAT, APPROVIZIONARE, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA. The 'Reports' section lists All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports. The central workspace is titled 'Worksheet Query Builder' and contains the following SQL code:

```

1 CREATE TABLE REZERVARE (
2     cod_client NUMBER(5),
3     cod_camera NUMBER(4),
4     cod_pachet NUMBER(3),
5     data_sosire DATE DEFAULT SYSDATE,
6     pret NUMBER(5) NOT NULL,
7     CONSTRAINT pk_rezervare PRIMARY KEY (cod_client, cod_camera, cod_pachet, data_sosire),
8     CONSTRAINT fk_rezervare_client FOREIGN KEY (cod_client) REFERENCES CLIENT(cod_client),
9     CONSTRAINT fk_rezervare_camera FOREIGN KEY (cod_camera) REFERENCES CAMERA(cod_camera),
10    CONSTRAINT fk_rezervare_pachet FOREIGN KEY (cod_pachet) REFERENCES PACHET(cod_pachet)
11 );
12
13
14

```

Below the code, the 'Script Output' window displays the message 'Table REZERVARE created.' The status bar at the bottom right shows 'Line 6 Column 5 | Insert | Modified | Windows: C | 12:36 AM 5/19/2023'. The taskbar at the bottom of the screen includes icons for weather (16°C), search, file explorer, browser, and other applications.

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Victoriei', 150, 'Bucuresti', 'Romania');

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Unirii', 10, 'Craiova', 'Romania');

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Dunarii', 7, 'Dubova', 'Romania');

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Poiana Soarelui', 15, 'Brasov', 'Romania');

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Bulevardul Ferdinand', 10, 'Constanta', 'Romania');

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered) ANEXA ANGAJAT APPROVIZIONI CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page project

Worksheet Query Builder

```
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Victoriei', 150, 'Bucuresti', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Unirii', 10, 'Craiova', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Dunarii', 7, 'Dubova', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Poiana Soarelui', 15, 'Brasov', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Bulevardul Ferdinand', 10, 'Constanta', 'Romania');

select * from locatie;
```

Script Output X Query Result X

All Rows Fetched: 5 in 0.003 seconds

COD_LOCATIE	STRADA	NUMAR	ORAS	TARA
1	Calea Victoriei	150	Bucuresti	Romania
2	Calea Unirii	10	Craiova	Romania
3	Strada Dunarii	7	Dubova	Romania
4	Strada Poiana Soarelui	15	Brasov	Romania
5	Bulevardul Ferdinand	10	Constanta	Romania

Click on an identifier with the Control key down to perform "Go to Declaration"

USDGBP -0.72%

Line 7 Column 23 | Insert | Modified | Windows: C

8:47 PM 6/1/2023

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered) Views Indexes Packages Procedures Functions Operators Queues Queues Tables Triggers Types Sequences Materialized Views Materialized View Logs Synonyms Public Synonyms Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page project

Worksheet Query Builder

```
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Victoriei', 150, 'Bucuresti', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Unirii', 10, 'Craiova', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Dunarii', 7, 'Dubova', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'strada Poiana Soarelui', 15, 'Brasov', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Bulevardul Ferdinand', 10, 'Constanta', 'Romania');
```

Script Output X

Task completed in 0.033 seconds

1 row inserted.

1 row inserted.

1 row inserted.

Error starting at line : 7 in command -
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Poiana Soarelui', 15, 'Poiana Brasov', 'Romania')
Error report -
ORA-12899: value too large for column "C##PROJECT"."LOCATIE"."ORAS" (actual: 13, maximum: 10)

1 row inserted.

1 row inserted.

Line 5 Column 95 | Insert | Modified | Windows: C

16°C Mostly cloudy 9:50 AM 5/18/2023

ENG ☰ 5/18/2023

```

INSERT INTO HOTEL VALUES (chei.NEXTVAL, 1, 'Novotel',
TO_DATE('23-10-2019', 'DD-MM-YYYY'), 70, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 2, 'Ramada Plaza',
TO_DATE('21-08-2017', 'DD-MM-YYYY'), 40, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 3, 'Riviera',
TO_DATE('02-05-2016', 'DD-MM-YYYY'), 30, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 4, 'Alpin', TO_DATE('27-09-2012',
'DD-MM-YYYY'), 40, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 5, 'Vega', TO_DATE('15-03-2019',
'DD-MM-YYYY'), 60, 5);

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes a connection named 'grupu15_01g' and a project named 'project'. The 'Tables (Filtered)' section under the connection shows tables such as ANEXA, ANGAJAT, APPROBATION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA. The 'Reports' section lists various report types.

In the central 'Worksheet' pane, there is a script block containing the five INSERT statements shown above, followed by a SELECT statement:

```

INSERT INTO HOTEL VALUES (chei.NEXTVAL, 1, 'Novotel', TO_DATE('23-10-2019', 'DD-MM-YYYY'), 70, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 2, 'Ramada Plaza', TO_DATE('21-08-2017', 'DD-MM-YYYY'), 40, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 3, 'Riviera', TO_DATE('02-05-2016', 'DD-MM-YYYY'), 30, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 4, 'Alpin', TO_DATE('27-09-2012', 'DD-MM-YYYY'), 40, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 5, 'Vega', TO_DATE('15-03-2019', 'DD-MM-YYYY'), 60, 5);

select * from hotel;

```

The 'Query Result' pane at the bottom shows the output of the SELECT statement, displaying the following data:

COD_HOTEL	COD_LOCATIE	NUME	DATA_INFIINTARE	NR_CAMERE	STELE
1	6	1 Novotel	23-OCT-19	70	5
2	7	2 Ramada Plaza	21-AUG-17	40	4
3	8	3 Riviera	02-MAY-16	30	4
4	9	4 Alpin	27-SEP-12	40	5
5	10	5 Vega	15-MAR-19	60	5

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists 'grup151_01g' and 'project'. The 'Tables (Filtered)' section under 'project' includes ANEXA, ANGAJAT, APPROVIZION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA. The 'Reports' sidebar lists various report types. The 'Worksheet' tab contains the following SQL script:

```
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 1, 'Novotel', TO_DATE('23-10-2019', 'DD-MM-YYYY'), 70, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 2, 'Ramada Plaza', TO_DATE('21-08-2017', 'DD-MM-YYYY'), 40, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 3, 'Riviera', TO_DATE('02-05-2016', 'DD-MM-YYYY'), 30, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 4, 'Alpin', TO_DATE('27-09-2012', 'DD-MM-YYYY'), 40, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 5, 'Vega', TO_DATE('15-03-2019', 'DD-MM-YYYY'), 60, 5);

select * from hotel;

commit;
```

The 'Script Output' window shows the results of the execution:

```
1 row inserted.

Commit complete.
```

The status bar at the bottom indicates 'Task completed in 0.029 seconds'.

```
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Cif', 'Curatenie');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Lidl', 'Alimente');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Somnia', 'Textile');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Mobexpert', 'Mobilier');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Techfit', 'Echipament sportiv');
```

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered)

ANEXA ANGAJAT APPROVIZIONI CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Worksheet Query Builder

```
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Cif', 'Curatenie');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Lidl', 'Alimente');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Somnia', 'Textile');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Mobexpert', 'Mobilier');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Techfit', 'Echipament sportiv');

select * from partener;
```

Script Output X Query Result X

All Rows Fetched: 5 in 0.002 seconds

COD_PARTENER	NUME	TIP_PRODUSE
1	11 Cif	Curatenie
2	12 Lidl	Alimente
3	13 Somnia	Textile
4	14 Mobexpert	Mobilier
5	15 Techfit	Echipament sportiv

Click on an identifier with the Control key down to perform "Go to Declaration"

USDGBP -0.72% Search b Google Chrome Microsoft Edge VS Code Spotify

Line 7 Column 1 Insert Modified Windows: C 9:20 PM 6/1/2023

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered) Views Indexes Packages Procedures Functions Operators Queues Queues Tables Triggers Types Sequences Materialized Views Materialized View Logs Synonyms Public Synonyms Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Worksheet Query Builder

```
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Cif', 'Curatenie');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Lidl', 'Alimente');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Somnia', 'Textile');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Mobexpert', 'Mobilier');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Techfit', 'Echipament sportiv');
```

Script Output X

Task completed in 0.027 seconds

1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.

Line 5 Column 77 Insert Modified Windows: C 10:44 AM 5/18/2023

16°C Mostly cloudy Search b Google Chrome Microsoft Edge VS Code Spotify

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Popa', 'Florin', '5014103009878');

```

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Stoica', 'Roxana',
'2980408230012');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Enescu', 'Bogdan',
'1991016190779');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Diaconu', 'Laura',
'6010101013335');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Vasilescu', 'Andrei',
'1970305170569');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Dinu', 'Ana', '2880614110278');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Marin', 'Cristian',
'1960103150375');

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure under 'Connections' and 'Tables (Filtered)'. The central workspace contains a 'Worksheet' tab with the following SQL code:

```

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Popa', 'Florin', '5014103009878');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Stoica', 'Roxana', '2980408230012');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Enescu', 'Bogdan', '1991016190779');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Diaconu', 'Laura', '6010101013335');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Vasilescu', 'Andrei', '1970305170569');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Dinu', 'Ana', '2880614110278');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Marin', 'Cristian', '1960103150375');

select * from client;

```

Below the worksheet is a 'Script Output' tab showing the results of the query:

	COD_CLIENT	NUME	PRENUME	CNP
1	16	Popa	Florin	5014103009878
2	17	Stoica	Roxana	2980408230012
3	18	Enescu	Bogdan	1991016190779
4	19	Diaconu	Laura	6010101013335
5	20	Vasilescu	Andrei	1970305170569
6	21	Dinu	Ana	2880614110278
7	22	Marin	Cristian	1960103150375

The status bar at the bottom indicates the environment: 22°C Partly cloudy, Line 9 Column 1, Insert, Modified, Windows: C, ENG, 9:21 PM, 6/1/2023.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists a project connection with tables like ANEXA, ANGAJAT, APPROVIZIONARE, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PARTNER, REZERVARE, and TIP_CAMERA. The 'Worksheet' tab displays a script with multiple INSERT INTO statements for the CLIENT table, each followed by a success message: '1 row inserted.' The 'Script Output' tab shows the same messages. The bottom status bar indicates the script completed in 0.025 seconds at 4:55 PM on 5/18/2023.

```
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Popa', 'Florin', '5014103009878');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Stoica', 'Roxana', '2980408230012');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Enescu', 'Bogdan', '1991016190779');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Diaconu', 'Laura', '6010101013335');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Vasilescu', 'Andrei', '1970305170569');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Dinu', 'Ana', '2880614110278');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Marin', 'Cristian', '1960103150375');
```

```
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Single', 1, 20, 200);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Dubla', 2, 28, 350);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Apartament', 4, 60, 450);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Suite', 3, 50, 400);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Penthouse', 7, 120, 800);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Royal', 5, 80, 600);
```

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered) ANEXA ANGAJAT APPROVIZIONARE CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page Project

Worksheet Query Builder

```
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Single', 1, 20, 200);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Dubla', 2, 28, 350);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Apartment', 4, 60, 450);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Suite', 3, 50, 400);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Penthouse', 7, 120, 800);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Royal', 5, 80, 600);

select * from tip_camera;
```

Script Output x Query Result x

All Rows Fetched: 6 in 0.002 seconds

COD_TIP	NUME	CAPACITATE	SUPRAFATA	PRET
1	23 Single	1	20	200
2	24 Dubla	2	28	350
3	25 Apartment	4	60	450
4	26 Suite	3	50	400
5	27 Penthouse	7	120	800
6	28 Royal	5	80	600

Click on an identifier with the Control key down to perform "Go to Declaration"

Bulevardul Pipera Closed road Search ENG 9:22 PM 6/1/2023 Line 8 Column 1 Insert Modified Windows: C

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections project Tables (Filtered) ANEXA ANGAJAT APPROVIZIONARE CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Indexes Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page project

Worksheet Query Builder

```
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Single', 1, 20, 200);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Dubla', 2, 28, 350);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Apartment', 4, 60, 450);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Suite', 3, 50, 400);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Penthouse', 7, 120, 800);
INSERT INTO TIP_CAMERA_VALUES (chei.NEXTVAL, 'Royal', 5, 80, 600);
```

Script Output x

Task completed in 0.029 seconds

1 row inserted.
1 row inserted.

Click on an identifier with the Control key down to perform "Go to Declaration"

22°C Mostly cloudy Search ENG 6:20 PM 5/18/2023 Line 6 Column 6 Insert Modified Windows: C

```

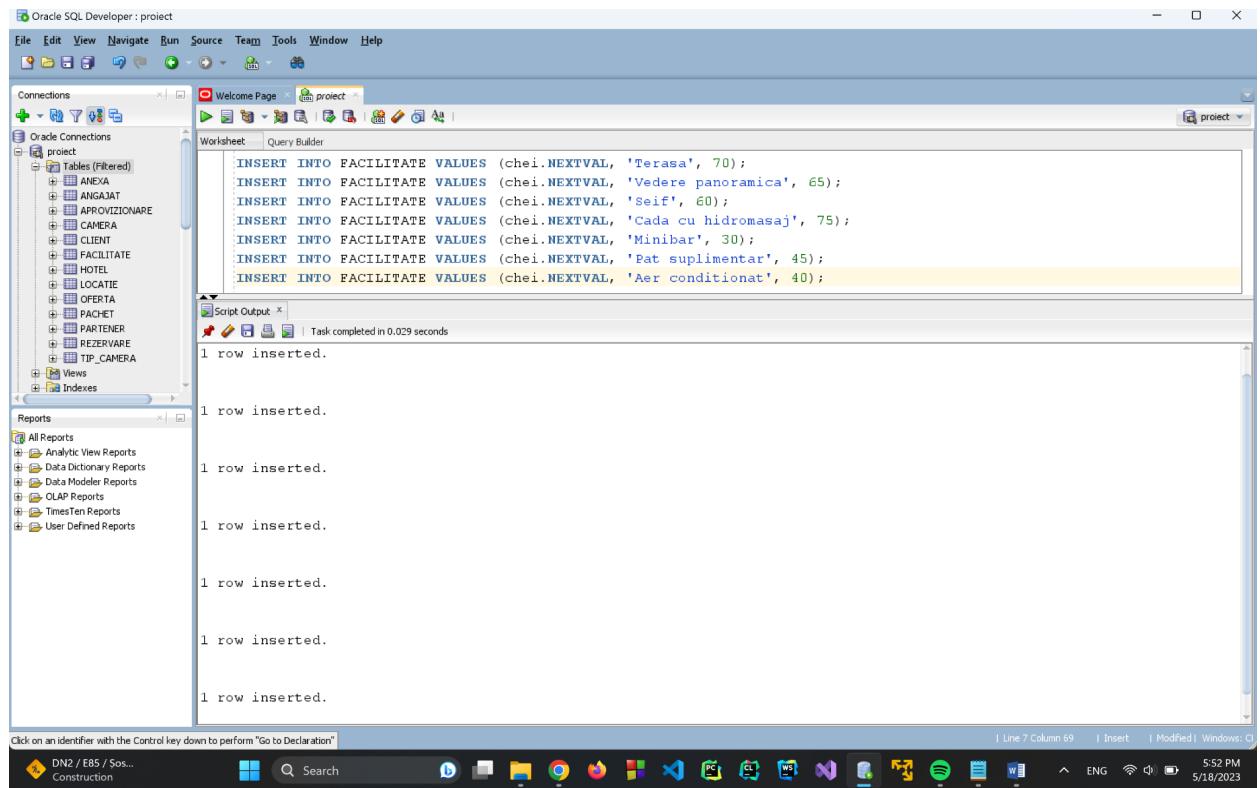
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Terasa', 70);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Vedere panoramica', 65);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Seif', 60);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Cada cu hidromasaj', 75);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Minibar', 30);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Pat suplimentar', 45);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Aer conditionat', 40);

```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays 'Connections' (Oracle Connections, grupa15_01g), 'Tables (Filtered)' (project containing ANEXA, ANGAJAT, APPROVIZION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, TIP_CAMERA, Views), and 'Reports' (All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports). The main workspace has tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab contains the SQL code shown above. Below it, the 'Script Output' tab shows the results of the query:

	COD_FACILITATE	FACILITATE	PRET
1	29	Terasa	70
2	30	Vedere panoramica	65
3	31	Seif	60
4	32	Cada cu hidromasaj	75
5	33	Minibar	30
6	34	Pat suplimentar	45
7	35	Aer conditionat	40

The status bar at the bottom indicates 'Click on an identifier with the Control key down to perform "Go to Declaration"', 'Line 9 Column 1', 'Insert', 'Modified', 'Windows: C:\', '22°C Partly cloudy', and the system clock '9:23 PM 6/1/2023'.



INSERT INTO OFERTA VALUES (27, 29);
INSERT INTO OFERTA VALUES (27, 30);
INSERT INTO OFERTA VALUES (27, 31);
INSERT INTO OFERTA VALUES (27, 32);
INSERT INTO OFERTA VALUES (27, 33);
INSERT INTO OFERTA VALUES (27, 34);
INSERT INTO OFERTA VALUES (27, 35);
INSERT INTO OFERTA VALUES (28, 29);
INSERT INTO OFERTA VALUES (28, 30);
INSERT INTO OFERTA VALUES (28, 32);
INSERT INTO OFERTA VALUES (28, 33);
INSERT INTO OFERTA VALUES (28, 34);
INSERT INTO OFERTA VALUES (28, 35);
INSERT INTO OFERTA VALUES (25, 29);
INSERT INTO OFERTA VALUES (25, 30);
INSERT INTO OFERTA VALUES (25, 33);
INSERT INTO OFERTA VALUES (25, 34);
INSERT INTO OFERTA VALUES (25, 35);
INSERT INTO OFERTA VALUES (26, 30);
INSERT INTO OFERTA VALUES (26, 33);

```
INSERT INTO OFERTA VALUES (26, 34);
INSERT INTO OFERTA VALUES (26, 35);
INSERT INTO OFERTA VALUES (24, 30);
INSERT INTO OFERTA VALUES (24, 33);
INSERT INTO OFERTA VALUES (24, 35);
INSERT INTO OFERTA VALUES (23, 35);
```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure under 'Connections' and 'Tables (Filtered)'. The main workspace contains a 'Worksheet' tab with the following SQL code:

```
INSERT INTO OFERTA VALUES (26, 34);
INSERT INTO OFERTA VALUES (26, 35);
INSERT INTO OFERTA VALUES (24, 30);
INSERT INTO OFERTA VALUES (24, 33);
INSERT INTO OFERTA VALUES (24, 35);
INSERT INTO OFERTA VALUES (23, 35);

select * from oferta;
```

Below the worksheet is a 'Script Output' tab showing the results of the executed queries. The results are displayed in a table:

	COD_TIP	COD_FACILITATE
1	23	35
2	24	30
3	24	33
4	24	35
5	25	29
6	25	30
7	25	33
8	25	24

The status bar at the bottom indicates 'All Rows Fetched: 26 in 0.001 seconds'. The system tray shows the date and time as '6/1/2023 9:32 PM'.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists 'grupu15_01g' and a 'project'. The 'Tables (Filtered)' section under 'project' shows tables like ANEXA, ANGAJAT, APPROVIZION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVAR, and TIP_CAMERA. The 'Reports' sidebar lists various report types. The 'Worksheet' tab contains the following SQL code:

```

INSERT INTO OFERTA VALUES (27, 29);
INSERT INTO OFERTA VALUES (27, 30);
INSERT INTO OFERTA VALUES (27, 31);
INSERT INTO OFERTA VALUES (27, 32);
INSERT INTO OFERTA VALUES (27, 33);
INSERT INTO OFERTA VALUES (27, 34);
INSERT INTO OFERTA VALUES (27, 35);
INSERT INTO OFERTA VALUES (28, 29);
INSERT INTO OFERTA VALUES (28, 30);
INSERT INTO OFERTA VALUES (28, 32);
INSERT INTO OFERTA VALUES (28, 33);
INSERT INTO OFERTA VALUES (28, 34);
INSERT INTO OFERTA VALUES (28, 35);
INSERT INTO OFERTA VALUES (25, 29);
INSERT INTO OFERTA VALUES (25, 30);
INSERT INTO OFERTA VALUES (25, 33);
INSERT INTO OFERTA VALUES (25, 34);
INSERT INTO OFERTA VALUES (25, 35);
INSERT INTO OFERTA VALUES (26, 30);
INSERT INTO OFERTA VALUES (26, 33);
INSERT INTO OFERTA VALUES (26, 34);
INSERT INTO OFERTA VALUES (26, 35);
INSERT INTO OFERTA VALUES (24, 30);
INSERT INTO OFERTA VALUES (24, 33);
INSERT INTO OFERTA VALUES (24, 35);
INSERT INTO OFERTA VALUES (23, 35);

select * from oferta;

```

Below the code, a note says "Click on an identifier with the Control key down to perform 'Go to Declaration'". The status bar at the bottom right shows "Line 28 Column 1 | Insert | Modified | Windows: C | 9:32 PM | ENG | 6/1/2023".

```

INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Parcare', 40);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Gradina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Teren de sport', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de conferinte', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de jocuri', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Spa', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de conferinte', 90);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Restaurant', 70);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de fitness', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Parcare', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de conferinte', 90);

```

```

INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Restaurant', 65);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Sala de fitness', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Parcare', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Restaurant', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de fitness', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de jocuri', 35);

```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying an SQL script. The script consists of several INSERT statements into the 'ANEXA' table and a final SELECT statement. The 'Script Output' tab shows the results of the execution, which is a table with columns: COD_ANEXA, COD_HOTEL, NUME, and CAPACITATE. The data is as follows:

COD_ANEXA	COD_HOTEL	NUME	CAPACITATE
1	36	10 Restaurant	95
2	37	10 Piscina	(null)
3	38	10 Sala de fitness	35
4	39	10 Spa	20
5	40	10 Parcare	40
6	41	10 Gradina	(null)
7	42	10 Teren de sport	(null)

The status bar at the bottom indicates the following information: Click on an identifier with the Control key down to perform "Go to Declaration". The system tray shows the date and time as 6/1/2023, 9:37 PM.

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : project". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has icons for New, Open, Save, Run, Stop, and Refresh. The Connections panel shows a connection named "grupu15_1_01g" under "Oracle Connections". The Projects panel shows a project named "project" containing tables like ANEXA, ANGAJAT, APPROVIZIONARE, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARAE, and TIP_CAMERA. The Reports panel lists various report types. The central workspace is titled "Worksheet" and contains a "Query Builder" tab. The query builder pane displays the following SQL code:

```

INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Parcare', 40);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Gradina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Teren de sport', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de conferinte', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de jocuri', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Spa', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de conferinte', 90);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Restaurant', 70);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de fitness', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Parcare', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de conferinte', 90);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Restaurant', 65);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Sala de fitness', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Parcare', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Restaurant', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de fitness', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de jocuri', 35);

```

Below the code, a note says "Click on an identifier with the Control key down to perform 'Go to Declaration'". The status bar at the bottom right shows "Line 9 Column 67 | Insert | Modified | Windows: C | 9:37 PM | 6/1/2023". The system tray shows the date and time as "22°C Partly cloudy" and "9:37 PM 6/1/2023".

```

INSERT INTO APROVIZIONARE VALUES (11, 10, TO_DATE('23-04-2023',
'DD-MM-YYYY'), 1000);
INSERT INTO APROVIZIONARE VALUES (12, 10, TO_DATE('28-05-2023',
'DD-MM-YYYY'), 1500);
INSERT INTO APROVIZIONARE VALUES (13, 10, TO_DATE('05-05-2023',
'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (14, 10, TO_DATE('01-03-2023',
'DD-MM-YYYY'), 4000);
INSERT INTO APROVIZIONARE VALUES (15, 10, TO_DATE('30-01-2023',
'DD-MM-YYYY'), 1500);
INSERT INTO APROVIZIONARE VALUES (11, 6, TO_DATE('21-04-2023',
'DD-MM-YYYY'), 2000);
INSERT INTO APROVIZIONARE VALUES (12, 6, TO_DATE('24-05-2023',
'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (13, 6, TO_DATE('09-05-2023',
'DD-MM-YYYY'), 3000);
INSERT INTO APROVIZIONARE VALUES (14, 6, TO_DATE('01-03-2022',
'DD-MM-YYYY'), 5000);

```

```
    INSERT INTO APROVIZIONARE VALUES (15, 6, TO_DATE('25-02-2023',
'DD-MM-YYYY'), 2000);
    INSERT INTO APROVIZIONARE VALUES (11, 7, TO_DATE('19-04-2023',
'DD-MM-YYYY'), 900);
    INSERT INTO APROVIZIONARE VALUES (12, 7, TO_DATE('28-05-2023',
'DD-MM-YYYY'), 1100);
    INSERT INTO APROVIZIONARE VALUES (13, 7, TO_DATE('06-03-2023',
'DD-MM-YYYY'), 1800);
    INSERT INTO APROVIZIONARE VALUES (15, 7, TO_DATE('12-01-2023',
'DD-MM-YYYY'), 800);
    INSERT INTO APROVIZIONARE VALUES (11, 8, TO_DATE('26-04-2023',
'DD-MM-YYYY'), 1200);
    INSERT INTO APROVIZIONARE VALUES (12, 8, TO_DATE('22-05-2023',
'DD-MM-YYYY'), 1700);
    INSERT INTO APROVIZIONARE VALUES (13, 8, TO_DATE('05-02-2023',
'DD-MM-YYYY'), 1400);
    INSERT INTO APROVIZIONARE VALUES (15, 8, TO_DATE('29-01-2023',
'DD-MM-YYYY'), 500);
    INSERT INTO APROVIZIONARE VALUES (11, 9, TO_DATE('24-04-2023',
'DD-MM-YYYY'), 1300);
    INSERT INTO APROVIZIONARE VALUES (12, 9, TO_DATE('27-05-2023',
'DD-MM-YYYY'), 1600);
    INSERT INTO APROVIZIONARE VALUES (13, 9, TO_DATE('02-05-2023',
'DD-MM-YYYY'), 2300);
    INSERT INTO APROVIZIONARE VALUES (14, 9, TO_DATE('08-03-2022',
'DD-MM-YYYY'), 1900);
    INSERT INTO APROVIZIONARE VALUES (15, 9, TO_DATE('31-01-2023',
'DD-MM-YYYY'), 1200);
```

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections grupa151_01g project Tables (Filtered) ANEXA ANGAJAT APROVIZIONARE CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARA TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page Project

Worksheet Query Builder

```
INSERT INTO APROVIZIONARE VALUES (12, 6, TO_DATE('24-05-2023', 'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (13, 6, TO_DATE('09-05-2023', 'DD-MM-YYYY'), 3000);
INSERT INTO APROVIZIONARE VALUES (14, 6, TO_DATE('01-03-2022', 'DD-MM-YYYY'), 5000);
INSERT INTO APROVIZIONARE VALUES (15, 6, TO_DATE('25-02-2023', 'DD-MM-YYYY'), 2000);

INSERT INTO APROVIZIONARE VALUES (11, 7, TO_DATE('19-04-2023', 'DD-MM-YYYY'), 900);
INSERT INTO APROVIZIONARE VALUES (12, 7, TO_DATE('28-05-2023', 'DD-MM-YYYY'), 1100);
INSERT INTO APROVIZIONARE VALUES (13, 7, TO_DATE('06-03-2023', 'DD-MM-YYYY'), 1800);
INSERT INTO APROVIZIONARE VALUES (15, 7, TO_DATE('12-01-2023', 'DD-MM-YYYY'), 800);

INSERT INTO APROVIZIONARE VALUES (11, 8, TO_DATE('26-04-2023', 'DD-MM-YYYY'), 1200);
INSERT INTO APROVIZIONARE VALUES (12, 8, TO_DATE('22-05-2023', 'DD-MM-YYYY'), 1700);
INSERT INTO APROVIZIONARE VALUES (13, 8, TO_DATE('05-02-2023', 'DD-MM-YYYY'), 1400);
INSERT INTO APROVIZIONARE VALUES (15, 8, TO_DATE('29-01-2023', 'DD-MM-YYYY'), 500);

INSERT INTO APROVIZIONARE VALUES (11, 9, TO_DATE('24-04-2023', 'DD-MM-YYYY'), 1300);
INSERT INTO APROVIZIONARE VALUES (12, 9, TO_DATE('27-05-2023', 'DD-MM-YYYY'), 1600);
INSERT INTO APROVIZIONARE VALUES (13, 9, TO_DATE('02-05-2023', 'DD-MM-YYYY'), 2300);
INSERT INTO APROVIZIONARE VALUES (14, 9, TO_DATE('08-03-2022', 'DD-MM-YYYY'), 1900);
INSERT INTO APROVIZIONARE VALUES (15, 9, TO_DATE('31-01-2023', 'DD-MM-YYYY'), 1200);

select * from aprovizacionare;
```

Script Output | Query Result

COD_PARTENER	COD_HOTEL	DATA_APROVIZIONARE	COST
1	11	10 23-APR-23	1000
2	12	10 28-MAY-23	1500
3	13	10 05-MAY-23	2500
4	14	10 01-MAR-23	4000
5	15	10 30-JAN-23	1500

Click on an identifier with the Control key down to perform "Go to Declaration"

22°C Partly cloudy

Line 29 Column 1 Insert Modified Windows: C

9:44 PM 6/1/2023

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections grupa151_01g project Tables (Filtered) ANEXA ANGAJAT APROVIZIONARE CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARA TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Welcome Page Project

Worksheet Query Builder

```
INSERT INTO APROVIZIONARE VALUES (11, 10, TO_DATE('23-04-2023', 'DD-MM-YYYY'), 1000);
INSERT INTO APROVIZIONARE VALUES (12, 10, TO_DATE('28-05-2023', 'DD-MM-YYYY'), 1500);
INSERT INTO APROVIZIONARE VALUES (13, 10, TO_DATE('05-05-2023', 'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (14, 10, TO_DATE('01-03-2023', 'DD-MM-YYYY'), 4000);
INSERT INTO APROVIZIONARE VALUES (15, 10, TO_DATE('30-01-2023', 'DD-MM-YYYY'), 1500);
INSERT INTO APROVIZIONARE VALUES (11, 6, TO_DATE('21-04-2023', 'DD-MM-YYYY'), 2000);
INSERT INTO APROVIZIONARE VALUES (12, 6, TO_DATE('24-05-2023', 'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (13, 6, TO_DATE('09-05-2023', 'DD-MM-YYYY'), 3000);
INSERT INTO APROVIZIONARE VALUES (14, 6, TO_DATE('01-03-2022', 'DD-MM-YYYY'), 5000);
INSERT INTO APROVIZIONARE VALUES (15, 6, TO_DATE('25-02-2023', 'DD-MM-YYYY'), 2000);
INSERT INTO APROVIZIONARE VALUES (11, 7, TO_DATE('19-04-2023', 'DD-MM-YYYY'), 900);
INSERT INTO APROVIZIONARE VALUES (12, 7, TO_DATE('28-05-2023', 'DD-MM-YYYY'), 1100);
INSERT INTO APROVIZIONARE VALUES (13, 7, TO_DATE('06-03-2023', 'DD-MM-YYYY'), 1800);
INSERT INTO APROVIZIONARE VALUES (15, 7, TO_DATE('12-01-2023', 'DD-MM-YYYY'), 800);
INSERT INTO APROVIZIONARE VALUES (11, 8, TO_DATE('26-04-2023', 'DD-MM-YYYY'), 1200);
INSERT INTO APROVIZIONARE VALUES (12, 8, TO_DATE('22-05-2023', 'DD-MM-YYYY'), 1700);
INSERT INTO APROVIZIONARE VALUES (13, 8, TO_DATE('05-02-2023', 'DD-MM-YYYY'), 1400);
INSERT INTO APROVIZIONARE VALUES (15, 8, TO_DATE('29-01-2023', 'DD-MM-YYYY'), 500);
INSERT INTO APROVIZIONARE VALUES (11, 9, TO_DATE('24-04-2023', 'DD-MM-YYYY'), 1300);
INSERT INTO APROVIZIONARE VALUES (12, 9, TO_DATE('27-05-2023', 'DD-MM-YYYY'), 1600);
INSERT INTO APROVIZIONARE VALUES (13, 9, TO_DATE('02-05-2023', 'DD-MM-YYYY'), 2300);
INSERT INTO APROVIZIONARE VALUES (14, 9, TO_DATE('08-03-2022', 'DD-MM-YYYY'), 1900);
INSERT INTO APROVIZIONARE VALUES (15, 9, TO_DATE('31-01-2023', 'DD-MM-YYYY'), 1200);
```

Click on an identifier with the Control key down to perform "Go to Declaration"

22°C Partly cloudy

Line 5 Column 86 Insert Modified Windows: C

9:44 PM 6/1/2023

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Constantinescu', 'Diana',
TO_DATE('27-05-1988', 'DD-MM-YYYY'), 'Manager', 35000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Pop', 'Adrian',
TO_DATE('18-08-1990', 'DD-MM-YYYY'), 'Manager', 30000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionescu', 'Radu',
TO_DATE('15-07-1991', 'DD-MM-YYYY'), 'Manager', 33000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Popescu', 'Alexandru',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 32000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Radulescu', 'Luca',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 31000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Marinescu', 'Andrei',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 18000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Valeriu', 'George',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 22000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Radu', 'Mihai',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 27000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Nistor', 'David',
TO_DATE('09/04/1988', 'DD-MM-YYYY'), 'Receptioner', 6500, 65);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Calinescu', 'Cosmin',
TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 6100, 65);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Zaharia', 'Marius',
TO_DATE('16/11/1987', 'DD-MM-YYYY'), 'Bucatar', 8500, 67);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Toma', 'Andrei',
TO_DATE('07/10/1997', 'DD-MM-YYYY'), 'Chelner', 4780, 67);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Stanciu', 'Stefan',
TO_DATE('25/02/1991', 'DD-MM-YYYY'), 'Chelner', 4900, 67);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Munteanu', 'Mihai',
TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 5200, 67);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Neagu', 'Oana',
TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3975, 66);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Cojocaru', 'Cosmina',
TO_DATE('18/06/1995', 'DD-MM-YYYY'), 'Menajera', 3950, 66);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionita', 'Irina',
TO_DATE('24/08/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 66);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Avram', 'Cristian',
TO_DATE('30/07/1985', 'DD-MM-YYYY'), 'Receptioner', 5900, 68);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Puiu', 'Mihnea',
TO_DATE('21/01/1982', 'DD-MM-YYYY'), 'Receptioner', 5750, 68);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Albu', 'Victor',
TO_DATE('01/08/1983', 'DD-MM-YYYY'), 'Bucatar', 8000, 68);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Florea', 'Marian',
TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Chelner', 4300, 69);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Tudor', 'David',
TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 4350, 69);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Petrescu', 'Bianca',
TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 3600, 69);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Iacob', 'Giorgiana',
TO_DATE('29/12/1991', 'DD-MM-YYYY'), 'Menajera', 3800, 69);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Gavrila', 'Mihai',
TO_DATE('20/01/1987', 'DD-MM-YYYY'), 'Receptioner', 5200, 70);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Ghinea', 'Mihnea',
TO_DATE('22/02/1997', 'DD-MM-YYYY'), 'Bucatar', 6500, 70);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Vasile', 'Tudor',
TO_DATE('25/07/1995', 'DD-MM-YYYY'), 'Chelner', 3800, 70);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Mihai', 'Alexandra',
TO_DATE('12/04/1983', 'DD-MM-YYYY'), 'Menajera', 3400, 70);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Iovan', 'Gabriel',
TO_DATE('17/08/1989', 'DD-MM-YYYY'), 'Receptioner', 5350, 71);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Gheorghe', 'Roxana',
TO_DATE('26/09/1981', 'DD-MM-YYYY'), 'Bucatar', 7000, 71);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Hristea', 'Sebastian',
TO_DATE('12/05/1990', 'DD-MM-YYYY'), 'Chelner', 3900, 71);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Stan', 'Alexia',
TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Menajera', 3800, 71);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Anghelina', 'George',
TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 5500, 72);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Sandu', 'Alin',
TO_DATE('07/05/1980', 'DD-MM-YYYY'), 'Bucatar', 7500, 72);

```

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Cristea', 'Dumitru',
TO_DATE('08/06/1984', 'DD-MM-YYYY'), 'Chelner', 4000, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Dumitrescu', 'Raluca',
TO_DATE('26/12/1982', 'DD-MM-YYYY'), 'Chelner', 4100, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Stanescu', 'Ana',
TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 4000, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Enache', 'Corina',
TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 72);

```

The screenshot shows the Oracle SQL Developer interface. The Worksheet pane contains a large block of SQL code for inserting data into the ANGAJAT table. The Query Result pane displays the results of the executed query, showing five rows of data.

```

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Tudor', 'David', TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 4350, 69)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Petrescu', 'Bianca', TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 3600, 70)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Iacob', 'Giorgiana', TO_DATE('29/12/1991', 'DD-MM-YYYY'), 'Menajera', 3800, 71)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Gavrilă', 'Mihai', TO_DATE('20/01/1987', 'DD-MM-YYYY'), 'Receptioner', 520)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Ghinea', 'Mihnea', TO_DATE('12/02/1997', 'DD-MM-YYYY'), 'Bucatar', 6500, 72)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Vasile', 'Tudor', TO_DATE('25/07/1995', 'DD-MM-YYYY'), 'Chelner', 3800, 70)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Mihai', 'Alexandra', TO_DATE('12/04/1983', 'DD-MM-YYYY'), 'Menajera', 3400, 71)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Iovan', 'Gabriel', TO_DATE('17/08/1989', 'DD-MM-YYYY'), 'Receptioner', 535)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Gheorghe', 'Roxana', TO_DATE('26/09/1981', 'DD-MM-YYYY'), 'Bucatar', 7000, 72)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Hristea', 'Sebastian', TO_DATE('12/05/1990', 'DD-MM-YYYY'), 'Chelner', 390)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Stan', 'Alexia', TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Menajera', 3800, 71)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Anghelina', 'George', TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 540)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Sandu', 'Alin', TO_DATE('07/05/1980', 'DD-MM-YYYY'), 'Bucatar', 7500, 72)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Cristea', 'Dumitru', TO_DATE('08/06/1984', 'DD-MM-YYYY'), 'Chelner', 4000, 70)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Dumitrescu', 'Raluca', TO_DATE('26/12/1982', 'DD-MM-YYYY'), 'Chelner', 4100, 70)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Stanescu', 'Ana', TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 4000, 70)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Enache', 'Corina', TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 70)

```

COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
1	65	10 Constantinescu	Diana	27-MAY-88	Manager	35000	(null)
2	66	10 Pop	Adrian	18-AUG-90	Manager	30000	(null)
3	67	10 Ionescu	Radu	15-JUL-91	Manager	33000	(null)
4	68	6 Popescu	Alexandru	27-MAY-90	Manager	32000	(null)
5	69	6 Radulescu	Luca	27-MAY-90	Manager	31000	(null)

```

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Constantinescu', 'Diana', TO_DATE('27-05-1988', 'DD-MM-YYYY'), 'Manager', 30000, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Pop', 'Adrian', TO_DATE('18-08-1990', 'DD-MM-YYYY'), 'Manager', 30000, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionescu', 'Radu', TO_DATE('15-07-1991', 'DD-MM-YYYY'), 'Manager', 33000, NULL)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Popescu', 'Alexandru', TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 32000, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Radulescu', 'Luca', TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 31000, NULL)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Marinescu', 'Andrei', TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 18000, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Valeriu', 'George', TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 22000, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Radu', 'Mihai', TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 27000, NULL)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Nistor', 'David', TO_DATE('09/04/1988', 'DD-MM-YYYY'), 'Receptioner', 6500, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Calinescu', 'Cosmin', TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 6500, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Zaharia', 'Marius', TO_DATE('16/11/1987', 'DD-MM-YYYY'), 'Bucatar', 8500, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Toma', 'Andrei', TO_DATE('07/10/1997', 'DD-MM-YYYY'), 'Chelner', 4780, 67)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Stanciu', 'Stefan', TO_DATE('25/02/1991', 'DD-MM-YYYY'), 'Chelner', 4900, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Munteanu', 'Mihai', TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 5200, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Neagu', 'Oana', TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3975, 66)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Cojocaru', 'Cosmina', TO_DATE('18/06/1995', 'DD-MM-YYYY'), 'Menajera', 3500, 65)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionita', 'Irina', TO_DATE('24/08/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 65)

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Avram', 'Cristian', TO_DATE('30/07/1985', 'DD-MM-YYYY'), 'Receptioner', 5500, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Puiu', 'Mihnea', TO_DATE('21/01/1982', 'DD-MM-YYYY'), 'Receptioner', 5750, NULL)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Albu', 'Victor', TO_DATE('01/08/1983', 'DD-MM-YYYY'), 'Bucatar', 8000, 68)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Florea', 'Marian', TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Chelner', 4300, 65)
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Tudor', 'David', TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 4350, 65)

```

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 80, 8);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 72, 7);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 55, 5);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 63, 6);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 61, 6);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 20, 2);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 28, 2);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 29, 2);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 22, 2);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 18, 1);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 14, 1);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 13, 1);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 12, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 27, 66, 6);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 54, 5);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 56, 5);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 44, 4);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 38, 3);
 INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 41, 4);

```
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 27, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 23, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 21, 2);
```

```
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 27, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 28, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 25, 32, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 48, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 24, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 17, 1);
```

```
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 25, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 31, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 35, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 21, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 17, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 10, 1);
```

```
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 25, 34, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 12, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 11, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 10, 1);
```

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections grupa151_01g project Tables (Filtered) ANEXA ANGAJAT APPROVIZIONI CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Worksheet Query Builder

```

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 24, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 17, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 25, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 31, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 35, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 21, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 17, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 10, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 25, 34, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 12, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 11, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 10, 1);

select * from camera;

```

Script Output X Query Result X

All Rows Fetched: 46 in 0.009 seconds

	COD_CAMERA	COD_HOTEL	COD_TIP	NUMAR	ETAJ
1	103	10	27	80	8
2	104	10	27	72	7
3	105	10	28	55	5
4	106	10	28	63	6
5	107	10	28	61	6
6	108	10	25	20	2
7	109	10	25	28	2

Click on an identifier with the Control key down to perform "Go to Declaration"

22°C Mostly cloudy

Line 52 Column 1 Insert Modified Windows: C

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections grupa151_01g project Tables (Filtered) ANEXA ANGAJAT APPROVIZIONI CAMERA CLIENT FACILITATE HOTEL LOCATIE OFERTA PACHET PARTENER REZERVARE TIP_CAMERA Views Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Worksheet Query Builder

```

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 80, 8);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 72, 7);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 55, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 63, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 61, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 20, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 28, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 18, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 14, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 12, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 27, 66, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 54, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 38, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 27, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 23, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 21, 2);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 27, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 28, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 25, 32, 3);
TRUNCATE TABLE CAMERA_VALUES (chei.NEXTVAL)

```

Click on an identifier with the Control key down to perform "Go to Declaration"

22°C Mostly cloudy

Line 51 Column 1 Insert Modified Windows: C

```

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 27, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 23, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 21, 2);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 27, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 28, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 25, 32, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 48, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 24, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 17, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 25, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 31, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 35, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 21, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 17, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 10, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 25, 34, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 12, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 11, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 10, 1);

```

INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'All Inclusive', 500);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Spa si Sala de fitness', 300);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Piscina', 250);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Teren de sport si Sala de jocuri', 200);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Tur ghidat', 150);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Activitati nautice', 300);

 INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'All Inclusive', 500);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Spa si Sala de fitness', 350);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Piscina', 230);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la muzeele din oras', 200);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Tur ghidat', 170);

 INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'All Inclusive', 400);
 INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la muzeele din oras', 140);

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la Piscina', 150);
```

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'All Inclusive', 400);
```

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'Acces la Piscina', 180);
```

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'All Inclusive', 450);
```

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Drumetii pe munte', 350);
```

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Spa si Sala de fitness',
```

275);

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Sala de jocuri', 120);
```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure under 'Connections' and 'Tables (Filtered)'. The 'Worksheet' tab is active, containing the following SQL code:

```
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'All Inclusive', 500);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Spa si Sala de fitness', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Piscina', 230);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la muzeele din oras', 200);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Tur ghidat', 170);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la muzeele din oras', 140);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la Piscina', 150);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'Acces la Piscina', 180);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'All Inclusive', 450);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Drumetii pe munte', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Spa si Sala de fitness', 275);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Sala de jocuri', 120);

select * from pachet;
```

The 'Query Result' tab shows the output of the last query:

COD_PACHET	COD_HOTEL	BENEFICIU	PRET
1	149	10 All Inclusive	500
2	150	10 Acces la Spa si Sala de fitness	300
3	151	10 Acces la Piscina	250
4	152	10 Acces la Teren de sport si Sala de jocuri	200
5	153	10 Tur ghidat	150
6	154	10 Activitati nautice	300
7	155	6 All Inclusive	500
8	156	6 Acces la Spa si Sala de fitness	250

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes 'Oracle Connections' (group15_1_01g) and a 'project' node containing tables like ANEXA, ANGAJAT, APPROVIZION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA. The 'Reports' section lists various report types. The main workspace is titled 'Worksheet' and contains a 'Query Builder' tab. The query editor displays several INSERT INTO statements for the 'PACHET' table, specifically targeting the 'chei.NEXTVAL' column. The statements are as follows:

```

INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'All Inclusive', 500);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Spa si Sala de fitness', 300);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Piscina', 250);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Teren de sport si Sala de jocuri', 200);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Tur ghidat', 150);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Activitati nautice', 300);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'All Inclusive', 500);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Spa si Sala de fitness', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Piscina', 230);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la muzeele din oras', 200);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Tur ghidat', 170);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la muzeele din oras', 140);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la Piscina', 150);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'Acces la Piscina', 180);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'All Inclusive', 450);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Drumetii pe munte', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Spa si Sala de fitness', 275);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Sala de jocuri', 120);

```

Below the code editor, the status bar shows 'Click on an identifier with the Control key down to perform "Go to Declaration"', 'Line 24 Column 77', 'Insert', 'Modified', 'Windows: C', and the system tray includes icons for weather (22°C, Mostly cloudy), search, file, browser, and other system applications.

INSERT INTO REZERVARE VALUES (16, 136, 161, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 1100);

INSERT INTO REZERVARE VALUES (16, 109, 153, TO_DATE('21-04-2020', 'DD-MM-YYYY'), 1800);

INSERT INTO REZERVARE VALUES (20, 104, 149, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 2300);

INSERT INTO REZERVARE VALUES (20, 109, 150, TO_DATE('05-06-2020', 'DD-MM-YYYY'), 4400);

INSERT INTO REZERVARE VALUES (20, 128, 166, TO_DATE('09-08-2020', 'DD-MM-YYYY'), 5005);

INSERT INTO REZERVARE VALUES (20, 107, 149, TO_DATE('19-03-2021', 'DD-MM-YYYY'), 6700);

INSERT INTO REZERVARE VALUES (20, 115, 154, TO_DATE('27-10-2021', 'DD-MM-YYYY'), 3200);

INSERT INTO REZERVARE VALUES (17, 143, 164, TO_DATE('23-12-2020', 'DD-MM-YYYY'), 4500);

INSERT INTO REZERVARE VALUES (17, 138, 163, TO_DATE('08-02-2021', 'DD-MM-YYYY'), 3020);

```
INSERT INTO REZERVARE VALUES (17, 113, 149, TO_DATE('02-05-2021',  
'DD-MM-YYYY'), 4000);
```

```
INSERT INTO REZERVARE VALUES (17, 145, 164, TO_DATE('01-09-2021',  
'DD-MM-YYYY'), 1800);
```

```
INSERT INTO REZERVARE VALUES (18, 128, 166, TO_DATE('23-11-2019',  
'DD-MM-YYYY'), 2200);
```

```
INSERT INTO REZERVARE VALUES (18, 122, 160, TO_DATE('16-01-2021',  
'DD-MM-YYYY'), 1000);
```

```
INSERT INTO REZERVARE VALUES (21, 103, 149, TO_DATE('23-11-2019',  
'DD-MM-YYYY'), 8000);
```

```
INSERT INTO REZERVARE VALUES (19, 131, 166, TO_DATE('23-10-2019',  
'DD-MM-YYYY'), 2200);
```

```
INSERT INTO REZERVARE VALUES (19, 106, 149, TO_DATE('24-08-2020',  
'DD-MM-YYYY'), 2000);
```

```
INSERT INTO REZERVARE VALUES (19, 123, 158, TO_DATE('27-07-2021',  
'DD-MM-YYYY'), 1800);
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays the 'Connections' tree, which includes 'grupa151_01g' and a 'project' node containing various tables like ANEXA, ANGAJAT, APPROVIZION, CAMERA, CLIENT, FACILITATE, HOTEL, LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA. Below the connections tree is a 'Reports' section with options like All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports.

The main workspace has tabs for 'Workshop' and 'Query Builder'. The 'Workshop' tab is active, showing a large block of SQL code for inserting data into the REZERVARE table. The code includes multiple INSERT statements with different values for COD_CLIENT, COD_CAMERA, COD_PACHET, DATA_SOSIRE, PRET, and various dates. At the bottom of the code area, there is a 'select * from rezervare;' statement.

The bottom part of the interface shows the 'Script Output' and 'Query Result' panes. The 'Query Result' pane displays the results of the last executed query, showing a table with columns COD_CLIENT, COD_CAMERA, COD_PACHET, DATA_SOSIRE, and PRET. The data is as follows:

	COD_CLIENT	COD_CAMERA	COD_PACHET	DATA_SOSIRE	PRET
1	16	136	161	23-NOV-19	1100
2	16	109	153	21-APR-20	1800
3	20	104	149	23-NOV-19	2300
4	20	109	150	05-JUN-20	4400
5	20	128	166	09-AUG-20	5005
6	20	107	149	19-MAR-21	6700
7	20	115	154	27-OCT-21	3200
8	20	110	154	20-DEC-20	1500

```

INSERT INTO REZERVARE VALUES (16, 136, 161, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 1100);
INSERT INTO REZERVARE VALUES (16, 109, 153, TO_DATE('21-04-2020', 'DD-MM-YYYY'), 1800);

INSERT INTO REZERVARE VALUES (20, 104, 149, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 2300);
INSERT INTO REZERVARE VALUES (20, 109, 150, TO_DATE('05-06-2020', 'DD-MM-YYYY'), 4400);
INSERT INTO REZERVARE VALUES (20, 128, 166, TO_DATE('09-08-2020', 'DD-MM-YYYY'), 5005);
INSERT INTO REZERVARE VALUES (20, 107, 149, TO_DATE('19-03-2021', 'DD-MM-YYYY'), 6700);
INSERT INTO REZERVARE VALUES (20, 115, 154, TO_DATE('27-10-2021', 'DD-MM-YYYY'), 3200);

INSERT INTO REZERVARE VALUES (17, 143, 164, TO_DATE('23-12-2020', 'DD-MM-YYYY'), 4500);
INSERT INTO REZERVARE VALUES (17, 138, 163, TO_DATE('08-02-2021', 'DD-MM-YYYY'), 3020);
INSERT INTO REZERVARE VALUES (17, 113, 149, TO_DATE('02-05-2021', 'DD-MM-YYYY'), 4000);
INSERT INTO REZERVARE VALUES (17, 145, 164, TO_DATE('01-09-2021', 'DD-MM-YYYY'), 1800);

INSERT INTO REZERVARE VALUES (18, 128, 166, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 2200);
INSERT INTO REZERVARE VALUES (18, 122, 160, TO_DATE('16-01-2021', 'DD-MM-YYYY'), 1000);

INSERT INTO REZERVARE VALUES (21, 103, 149, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 8000);

INSERT INTO REZERVARE VALUES (19, 131, 166, TO_DATE('23-10-2019', 'DD-MM-YYYY'), 2200);
INSERT INTO REZERVARE VALUES (19, 106, 149, TO_DATE('24-08-2020', 'DD-MM-YYYY'), 2000);
INSERT INTO REZERVARE VALUES (19, 123, 158, TO_DATE('27-07-2021', 'DD-MM-YYYY'), 1800);

```

12. Formulati in limbaj natural si implementati 5 cereri SQL complexe ce vor utiliza, in ansamblul lor, urmatoarele elemente:

- subcereri sincronizate in care intervin cel putin 3 table
- subcereri nesincronizate in clauza FROM
- grupari de date cu subcereri nesincronizate in care intervin cel putin 3 table, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)
 - ordonari si utilizarea functiilor NVL si DECODE (in cadrul aceleiasi cereri)
 - utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE
 - utilizarea a cel putin 1 bloc de cerere (clauza WITH)

-- Sa se afiseze numele hotelurilor, anul infiintarii sale, numarul de luni care au trecut de atunci si gradul fiecaruia

-- de cautare(mic, mediu, mare). Aceasta va fi determinat in functie de numarul de camere rezervate din fiecare

-- hotel(sub 3 camere rezervate inseamna un grad de cautare scazut, intre 3 si 7 inseamna un grad de cautare mediu,

-- iar peste 7 inseamna un grad de cautare ridicat).

-- folosirea unei expresii CASE, a 2 functii pe date calendaristice si a unei subcereri nesincronizate in clauza FROM

```
SELECT h.nume AS nume_hotel, EXTRACT(YEAR FROM
TO_DATE(h.data_infiintare, 'DD-MM-YYYY')) An_infiintare,
FLOOR(MONTHS_BETWEEN(TO_DATE(SYSDATE, 'DD-MM-YYYY'),
TO_DATE(h.data_infiintare, 'DD-MM-YYYY'))) Luni_trecute,
CASE
    WHEN cnt_camere > 7 THEN 'mare'
    WHEN cnt_camere >= 3 AND cnt_camere <= 7 THEN 'mediu'
    ELSE 'mic'
END AS grad_cautare
FROM HOTEL h LEFT JOIN ( SELECT c.cod_hotel, COUNT(r.cod_camera) AS
cnt_camere
    FROM REZERVARE r JOIN CAMERA c ON (r.cod_camera =
c.cod_camera)
    GROUP BY c.cod_hotel
) sub ON (h.cod_hotel = sub.cod_hotel);
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar contains a Connections tree (project selected), Reports tree (All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports), and a Database schema browser (Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Triggers, Types, Sequences, Materialized View, Materialized View, Synonyms, Public Synonyms). The central workspace has tabs for Worksheet and Query Builder. The Worksheet tab displays the SQL query provided above. The Query Result tab shows the execution output:

NUME_HOTEL	AN_INFIINTARE	LUNI_TRECUTE	GRAD_CAUTARE
1 Novotel	19	42	mic
2 Ramada Plaza	17	69	mic
3 Riviera	16	84	mic
4 Alpin	12	127	mediu
5 Vega	19	50	mare

The status bar at the bottom indicates: Line 14 Column 22 | Insert | Modified | Windows: C | 19°C Sunny | 9:55 AM | 5/21/2023.

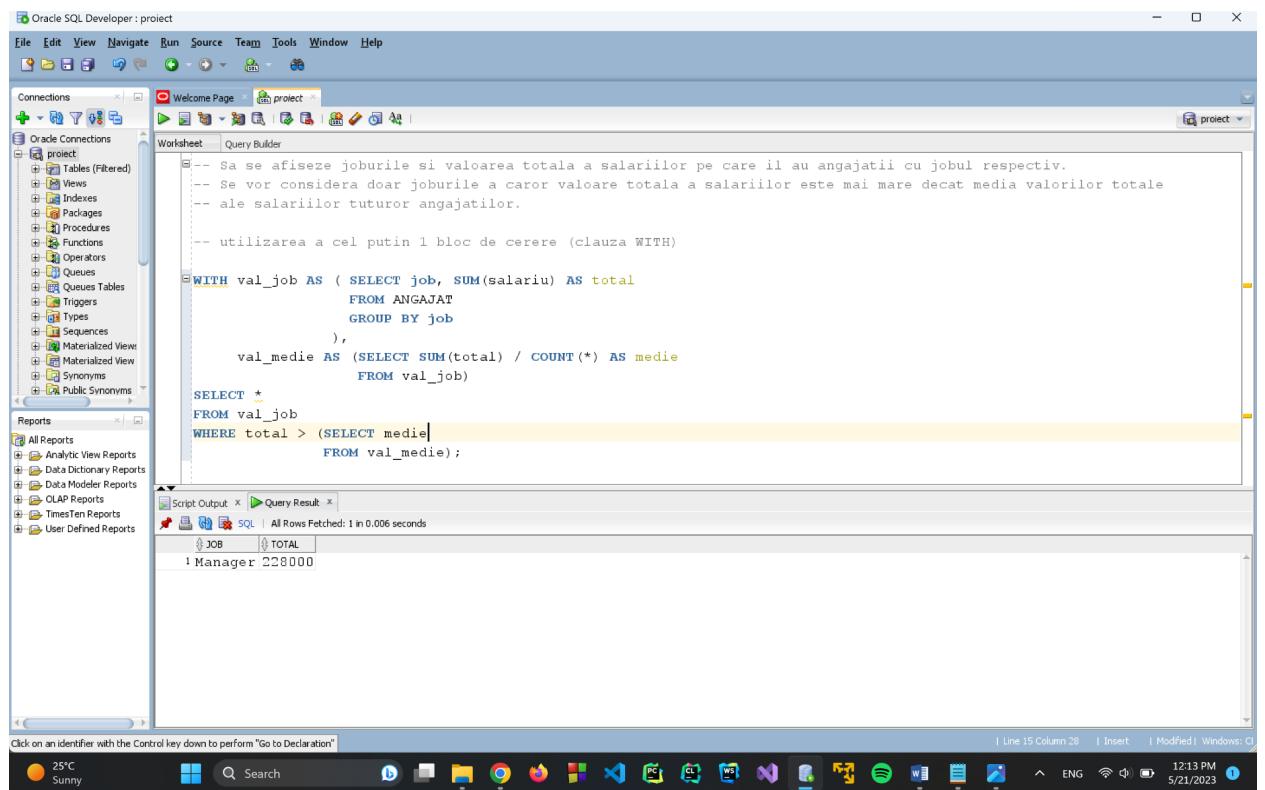
-- Sa se afiseze joburile si valoarea totala a salariilor pe care il au angajatii cu jobul respectiv.

-- Se vor considera doar joburile a caror valoare totala a salariilor este mai mare decat media valorilor totale

-- ale salariilor tuturor angajatilor.

-- utilizarea a cel putin 1 bloc de cerere (clauza WITH)

```
WITH val_job AS ( SELECT job, SUM(salariu) AS total
                   FROM ANGAJAT
                  GROUP BY job
                 ),
     val_medie AS (SELECT SUM(total) / COUNT(*) AS medie
                    FROM val_job)
SELECT *
  FROM val_job
 WHERE total > (SELECT medie
                  FROM val_medie);
```



-- Pentru fiecare hotel care a avut minim 5 aprovizionari, sa se afiseze numele hotelului si
-- suma capacitatilor tuturor anexelor, exluzand parcarile

-- ordonari si utilizarea functiilor NVL si DECODE (in cadrul aceleiasi cereri)
-- folosirea unei functii pe siruri de caractere - UPPER

```
SELECT h.nume, SUM(DECODE(UPPER(a.nume), 'PARCARE', 0, NVL(a.capacitate,  
0))) AS Capacitate  
FROM HOTEL h JOIN ANEXA a ON (h.cod_hotel = a.cod_hotel)  
WHERE h.cod_hotel IN ( SELECT cod_hotel  
                      FROM APROVIZIONARE  
                      GROUP BY cod_hotel  
                      HAVING COUNT(cod_partener) >= 5  
                    )  
GROUP BY h.nume, h.cod_hotel  
ORDER BY Capacitate DESC;
```

```

-- Pentru fiecare hotel care a avut minim 5 aprovizionari, sa se afiseze numele hotelului si
-- suma capacitatilor tuturor anexelor, exluzand parcarile

SELECT h.nume, SUM(DECODE(UPPER(a.nume), 'PARCARE', 0, NVL(a.capacitate, 0))) AS Capacitate
FROM HOTEL h JOIN ANEXA a ON (h.cod_hotel = a.cod_hotel)
WHERE h.cod_hotel IN ( SELECT cod_hotel
                        FROM APROVIZIONARE
                        GROUP BY cod_hotel
                        HAVING COUNT(cod_partener) >= 5
                      )
GROUP BY h.nume, h.cod_hotel
ORDER BY Capacitate DESC;

```

NUME	CAPACITATE
1 Vega	260
2 Novotel	245
3 Alpin	160

-- Sa se afiseze pentru fiecare client si pentru fiecare hotel la care acesta a rezervat macar o camera,

-- numele complet al clientului, in care numele de familie va fi scris cu majuscule, iar prenumele cu litere mici,

-- numele hotelului, pretul celei mai scumpe rezervari si numarul camerei la care a fost facuta rezervarea.

-- Rezultatele vor fi ordonate dupa numele complet

-- subcereri sincronizate in care intervin cel putin 3 table, folosirea a cel putin 2 functii -- pe siruri de caractere(CONCAT, UPPER, LOWER)

```
SELECT CONCAT(UPPER(c.nume) || ' ', LOWER(c.prenume)) Nume_complet, h.nume,
r.pret, cam.numar
```

```
FROM rezervare r JOIN CLIENT c on (r.cod_client = c.cod_client)
JOIN CAMERA cam on (r.cod_camera = cam.cod_camera)
JOIN HOTEL h on (cam.cod_hotel = h.cod_hotel)
```

```
WHERE r.pret = ( SELECT MAX(pret)
```

```

        FROM rezervare r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
        WHERE h1.cod_hotel = h.cod_hotel AND c.cod_client = c1.cod_client
    )
ORDER BY Nume_complet;

```

The screenshot shows the Oracle SQL Developer interface. The Worksheet tab contains a SQL query with comments explaining the logic. The Query Result tab displays the execution results.

```

-- Sa se afiseze pentru fiecare client si pentru fiecare hotel la care acesta a rezervat macar o camera,
-- numele complet al clientului, in care numele de familie va fi scris cu majuscula, iar prenumele cu litere mici,
-- numele hotelului, pretul celei mai scumpe rezervari si numarul camerei la care a fost facuta rezervarea.
-- Rezultatele vor fi ordonate dupa numele complet

-- subcereri sincronizate in care intervin cel putin 3 tabele, folosirea a cel putin 2 functii
-- pe siruri de caractere(CONCAT, UPPER, LOWER)

SELECT CONCAT(UPPER(c.nume) || ' ', LOWER(c.prenume)) Nume_complet, h.nume, r.pret, cam.numar
FROM rezervare r JOIN CLIENT c ON (r.cod_client = c.cod_client)
JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
WHERE r.pret = ( SELECT MAX(pret)
                  FROM rezervare r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                  JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                  JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                  WHERE h1.cod_hotel = h.cod_hotel AND c.cod_client = c1.cod_client
              )
ORDER BY Nume_complet;

```

NUME_COMPLET	NUME	PRET	NUMAR
1 DIACONU laura	Vega	2000	63
2 DIACONU laura	Alpin	2200	37
3 DIACONU laura	Novotel	1800	27
4 DINU ana	Vega	8000	80
5 ENESCU bogdan	Novotel	1000	37
6 ENESCU bogdan	Alpin	2200	44

-- Sa se afiseze numele hotelului cu cele mai multe incasari din rezervari in care cel putin 2 clienti sa fi facut o rezervare.

-- alaturi de suma totala obtinuta din incasarile respective

-- grupari de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)

```

SELECT h.nume, SUM(r.pret) AS Suma_totala
FROM rezervare r JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
GROUP BY h.cod_hotel, h.nume
HAVING COUNT(DISTINCT r.cod_client) >= 2 AND

```

```

SUM(r.pret) = ( SELECT MAX(sum_tot)
    FROM ( SELECT h1.cod_hotel, SUM(r1.pret) AS sum_tot
        FROM REZERVARE r1 JOIN CAMERA cam1 ON (r1.cod_camera =
cam1.cod_camera)
        JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
        GROUP BY h1.cod_hotel
    )
);

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** Oracle Connections - grupaISI_01ig project
- Worksheet:** Query Builder
 - SQL code (partially visible):


```
-- Sa se afiseze numele hotelului cu cele mai multe incasari din rezervari in care cel putin 2 clienti sa fi facut o rezervare
-- alaturi de suma totala obtinuta din incasarile respective

-- grupari de date cu subcereri nesincronizate in care intervin cel putin 3
-- tabele, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cerere)

SELECT h.numar, SUM(r.pret) AS Suma_totala
FROM rezervare r JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
GROUP BY h.cod_hotel, h.numar
HAVING COUNT(DISTINCT r.cod_client) >= 2 AND
       SUM(r.pret) = ( SELECT MAX(sum_tot)
                        FROM ( SELECT h1.cod_hotel, SUM(r1.pret) AS sum_tot
                            FROM REZERVARE r1 JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                            JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                            GROUP BY h1.cod_hotel
                        )
);

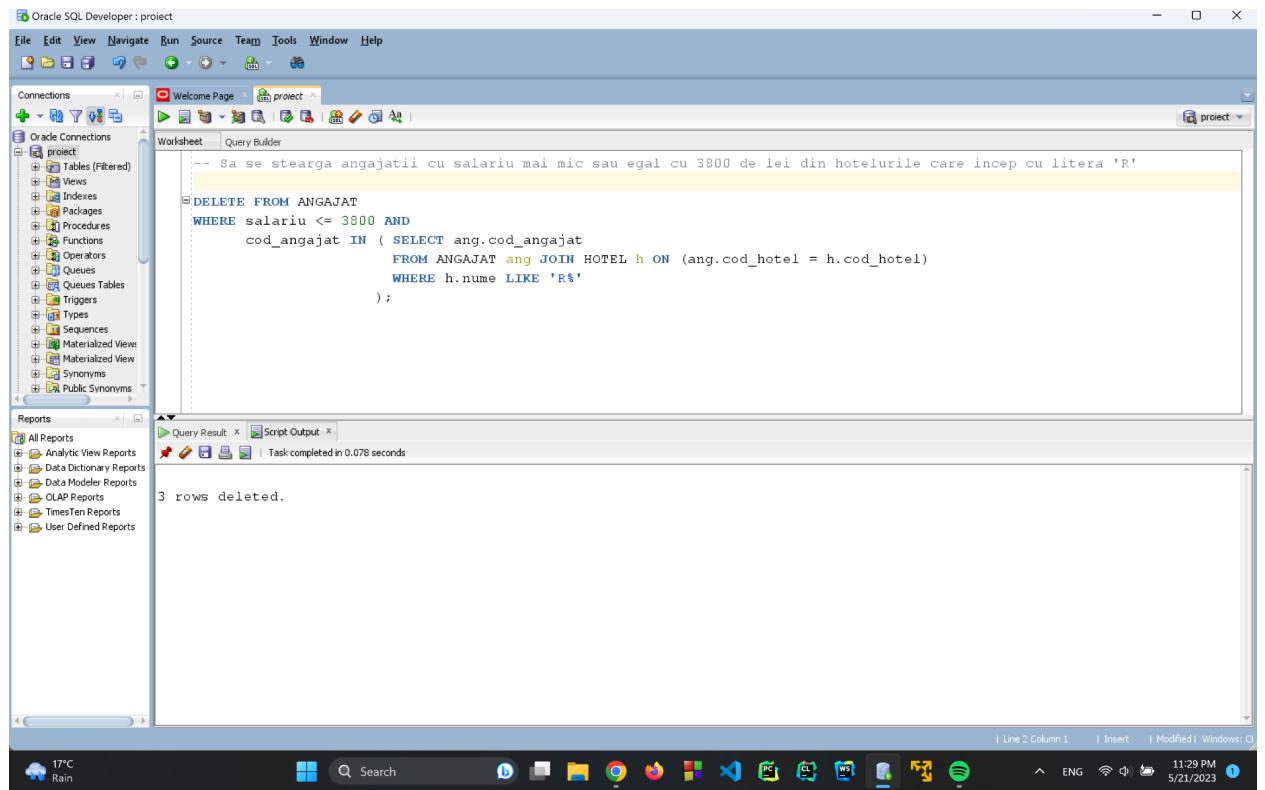
```
 - Query Result:**

NUMAR	SUMA_TOTALA
Vega	20400
- Bottom Status Bar:**
 - Cloudy weather icon: 22°C
 - Search icon
 - Browser icon
 - Google Chrome icon
 - File Explorer icon
 - Task View icon
 - Calculator icon
 - Power icon
 - Network icon
 - Volume icon
 - Language: ENG
 - Date/Time: 9:56 AM
 - Date: 6/1/2023

13. Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri.

-- Sa se stearga angajati cu salariu mai mic sau egal cu 3800 de lei din hotelurile care incep cu litera 'R'

```
DELETE FROM ANGAJAT
WHERE salariu <= 3800 AND
cod_angajat IN ( SELECT ang.cod_angajat
                  FROM ANGAJAT ang JOIN HOTEL h ON (ang.cod_hotel = h.cod_hotel)
                  WHERE h.nume LIKE 'R%'
                );
```



-- Sa se stearga rezervarile in care camera era de 2 persoane.

```
DELETE FROM REZERVARE
WHERE cod_camera IN ( SELECT cam.cod_camera
```

```

FROM rezervare r JOIN camera cam ON (r.cod_camera =
cam.cod_camera)
JOIN tip_camera t ON (cam.cod_tip = t.cod_tip)
WHERE t.capacitate = 2
);

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' and 'Reports' panes. The main workspace contains a 'Worksheet' tab with the following SQL code:

```

-- Sa se stearga rezervarile in care camera era de 2 persoane.

DELETE FROM REZERVARE
WHERE cod_camera IN ( SELECT cam.cod_camera
                      FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera)
                      JOIN tip_camera t ON (cam.cod_tip = t.cod_tip)
                     WHERE t.capacitate = 2
                   );

```

The 'Script Output' tab at the bottom shows the result of the execution:

```

3 rows deleted.

```

-- Sa se stearga toate aprovizionarile partenerului Cif in hotelurile de 4 stele

```

DELETE FROM APROVIZIONARE
WHERE (cod_partener, cod_hotel) IN ( SELECT p.cod_partener, h.cod_hotel
                                       FROM APROVIZIONARE a JOIN PARTENER p ON
                                         (p.cod_partener = a.cod_partener)
                                       JOIN HOTEL h ON (h.cod_hotel = a.cod_hotel)
                                       WHERE h.stele = 4 AND LOWER(p.nume) = 'cif'
                                     );

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists a project connection. The 'Worksheet' tab contains a SQL script:

```
-- Sa se stearga toate aprovizionarile partenerului Cif in hotelurile de 4 stele
DELETE FROM APROVIZIONARE
WHERE (cod_partener, cod_hotel) IN ( SELECT p.cod_partener, h.cod_hotel
                                      FROM APROVIZIONARE a JOIN PARTENER p ON (p.cod_partener = a.cod_partener)
                                      JOIN HOTEL h ON (h.cod_hotel = a.cod_hotel)
                                      WHERE h.stele = 4 AND LOWER(p.nume) = 'cif'
                                    );
```

The 'Script Output' window shows the result of the execution:

```
2 rows deleted.
```

The system tray at the bottom indicates it's 16°C with light rain, and the date and time are 5/21/2023 11:54 PM.

-- Actualizati salariile menajerelor cu media salariilor menajerelor din acelasi hotel

```
UPDATE ANGAJAT ang
SET ang.salariu = ( SELECT AVG(a.salariu)
                     FROM ANGAJAT a
                     WHERE a.job = 'Menajera' AND ang.cod_hotel = a.cod_hotel
                     GROUP BY a.cod_hotel
                   )
WHERE ang.job = 'Menajera';
```

```

-- Actualizati salariile menajerelor cu media salariilor menajerelor din acelasi hotel

UPDATE ANGAJAT ang
SET ang.salariu = ( SELECT AVG(a.salariu)
                     FROM ANGAJAT a
                     WHERE a.job = 'Menajera' AND ang.cod_hotel = a.cod_hotel
                     GROUP BY a.cod_hotel
                   )
WHERE ang.job = 'Menajera';

```

Script Output: Task completed in 0.048 seconds

7 rows updated.

-- Actualizati pretul rezervarii pentru rezervarile din lunile ianuarie si februarie a anului 2021 cu media pretului

-- platit pe rezervari a clientului cu cele mai multe rezervari

UPDATE REZERVARE

SET pret = (SELECT AVG(pret)

FROM REZERVARE

WHERE cod_client = (SELECT cod_client

FROM REZERVARE

GROUP BY cod_client

HAVING COUNT(*) = (SELECT MAX(nr_rez)

FROM (SELECT COUNT(*) AS nr_rez

FROM REZERVARE

GROUP BY cod_client

)

)

)

WHERE data_sosire >= TO_DATE('01-01-2021', 'DD-MM-YYYY') AND data_sosire < TO_DATE('01-03-2021', 'DD-MM-YYYY');

```

-- Actualizati pretul rezervarii pentru rezervarile din lunile ianuarie si februarie a anului 2021 cu media pretului
-- platit pe rezervari a clientului cu cele mai multe rezervari

UPDATE REZERVARE
SET pret = ( SELECT AVG(pret)
              FROM REZERVARE
            WHERE cod_client = ( SELECT cod_client
                                    FROM REZERVARE
                                    GROUP BY cod_client
                                    HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                            FROM ( SELECT COUNT(*) AS nr_rez
                                                    FROM REZERVARE
                                                    GROUP BY cod_client
                                              )
                                         )
                           )
            )
WHERE data_sosire >= TO_DATE('01-01-2021', 'DD-MM-YYYY') AND data_sosire < TO_DATE('01-03-2021', 'DD-MM-YYYY');

2 rows updated.

```

-- Actualizati salariile angajatilor ce nu sunt manageri, din hotelurile de 4 stele,
modificandu-le in salariul

-- angajatului care este nascut in acelasi an si in aceeasi luna cu managerul sau,
-- la care se adauga 500 de lei.

UPDATE ANGAJAT

```

SET salariu = ( SELECT ang.salariu + 500
                  FROM ANGAJAT ang JOIN ANGAJAT man ON (ang.cod_manager =
man.cod_angajat)
                  WHERE ang.cod_manager IS NOT NULL AND
                        EXTRACT(YEAR FROM ang.data_nasterii) = EXTRACT(YEAR FROM
man.data_nasterii) AND
                        EXTRACT(MONTH FROM ang.data_nasterii) = EXTRACT(MONTH
FROM man.data_nasterii)
                  )
WHERE cod_angajat IN ( SELECT ang.cod_angajat
                  FROM ANGAJAT ang JOIN HOTEL h ON (ang.cod_hotel = h.cod_hotel)
                  WHERE h.stele = 4 AND ang.cod_manager IS NOT NULL
                );

```

```

-- Actualizati salariile angajatilor ce nu sunt manageri, din hotelurile de 4 stele, modificandu-le in salariul
-- angajatului dintr-un hotel de 4 stele, care este nascut in acelasi an si in aceeasi luna cu managerul sau,
-- la care se adauga 500 de lei.

UPDATE ANGAJAT
SET salariu = ( SELECT ang.salariu + 500
                 FROM ANGAJAT ang JOIN ANGAJAT man ON (ang.cod_manager = man.cod_angajat)
                 WHERE ang.cod_manager IS NOT NULL AND
                       EXTRACT(YEAR FROM ang.data_nasterii) = EXTRACT(YEAR FROM man.data_nasterii) AND
                       EXTRACT(MONTH FROM ang.data_nasterii) = EXTRACT(MONTH FROM man.data_nasterii)
               )
WHERE cod_angajat IN ( SELECT ang.cod_angajat
                         FROM ANGAJAT ang JOIN HOTEL h ON (ang.cod_hotel = h.cod_hotel)
                         WHERE h.stele = 4 AND ang.cod_manager IS NOT NULL
                   );

```

Script Output X | Task completed in 0.031 seconds

5 rows updated.

14. Crearea unei vizualizari complexe. Dati un exemplu de operatie LMD permisa pe vizualizarea respectiva si un exemplu de operatie LMD nepermisa.
15. Formulati in limbaj natural si implementati in SQL: o cerere ce utilizeaza operatia outer-join pe minimum 4 tabele, o cerere ce utilizeaza operatia division si o cerere care implementeaza analiza top-n.

-- Sa se afiseze pentru toti clientii cu lungimea numelui cel putin 5: numele lor complet, pretul rezervarilor, numarul

-- camerei rezervate si numele hotelului la care au facut rezervarea, fiind inclusi si clientii care nu au rezervat

-- nicio camera la niciun hotel

-- outer-join pe minimum 4 tabele

```

SELECT c.nume || ' ' || c.prenume AS Nume_complet, r.pret, cam.numar, h.nume
FROM CLIENT c LEFT JOIN REZERVARE r ON (c.cod_client = r.cod_client)
LEFT JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
LEFT JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)

```

WHERE LENGTH(c.nume) >= 5;

```
-- Sa se afiseze pentru toti clientii cu lungimea numelui cel putin 5: numele lor complet, pretul rezervarilor, numarul
-- camerei rezervate si numele hotelului la care au facut rezervarea, fiind inclusi si clientii care nu au rezervat
-- nicio camera la niciun hotel

SELECT c.nume || ' ' || c.prenume AS Nume_complet, r.pret, cam.numar, h.numar
FROM CLIENT c LEFT JOIN REZERVARE r ON (c.cod_client = r.cod_client)
LEFT JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
LEFT JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
WHERE LENGTH(c.nume) >= 5;
```

NUME_COMPLET	PRET	NUMAR	NUME
1 Enescu Bogdan	1000	37	Novotel
2 Stoica Roxana	3020	35	Ramada Plaza
3 Stoica Roxana	4500	34	Riviera
4 Vasilescu Andrei	5005	44	Alpin
5 Enescu Bogdan	2200	44	Alpin
6 Diaconu Laura	2200	37	Alpin
7 Vasilescu Andrei	2300	72	Vega
8 Diaconu Laura	2000	63	Vega
9 Vasilescu Andrei	6700	61	Vega
10 Vasilescu Andrei	4400	28	Vega
11 Vasilescu Andrei	3200	12	Vega
12 Marin Cristian	(null)	(null)	(null)

-- Sa se afiseze numele clientilor care au facut rezervari la toate hotelurile la care s-a
cazat clientul

-- cu cele mai multe rezervari facute, incluzandu-l si pe clientul respectiv

-- division

```
SELECT c.nume || ' ' || c.prenume
FROM REZERVARE r JOIN CLIENT c on (r.cod_client = c.cod_client)
JOIN CAMERA cam on (r.cod_camera = cam.cod_camera)
JOIN HOTEL h on (cam.cod_hotel = h.cod_hotel)
WHERE h.cod_hotel IN ( SELECT h1.cod_hotel
FROM REZERVARE r1 JOIN CLIENT c1 on (r1.cod_client =
c1.cod_client)
JOIN CAMERA cam1 on (r1.cod_camera = cam1.cod_camera)
JOIN HOTEL h1 on (cam1.cod_hotel = h1.cod_hotel)
WHERE c1.cod_client = ( SELECT cod_client
FROM REZERVARE
GROUP BY cod_client
```

```

        HAVING COUNT(*) = (
            SELECT MAX(nr_rez)
            FROM (
                SELECT COUNT(*) AS nr_rez
                FROM REZERVARE
                GROUP BY cod_client
            )
        )
    )
GROUP BY c.nume, c.prenume
HAVING COUNT(DISTINCT h.cod_hotel) = (
    SELECT COUNT(DISTINCT
    h1.cod_hotel)
    FROM REZERVARE r1 JOIN CLIENT c1 on (r1.cod_client =
    c1.cod_client)
    JOIN CAMERA cam1 on (r1.cod_camera =
    cam1.cod_camera)
    JOIN HOTEL h1 on (cam1.cod_hotel = h1.cod_hotel)
    WHERE c1.cod_client = (
        SELECT cod_client
        FROM REZERVARE
        GROUP BY cod_client
        HAVING COUNT(*) = (
            SELECT MAX(nr_rez)
            FROM (
                SELECT COUNT(*) AS nr_rez
                FROM REZERVARE
                GROUP BY cod_client
            )
        )
    )
);

```

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page project

Worksheet Query Builder

```
-- Sa se afiseze numele clientilor care au facut rezervari la toate hotelurile la care s-a cazat clientul
-- cu cele mai multe rezervari facute, incluzandu-l si pe clientul respectiv

-- division

SELECT c.nume || ' ' || c.prenume
FROM REZERVARE r JOIN CLIENT c ON (r.cod_client = c.cod_client)
JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
WHERE h.cod_hotel IN ( SELECT h1.cod_hotel
                        FROM REZERVARE r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                        JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                        JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                        WHERE c1.cod_client = ( SELECT cod_client
                                                FROM REZERVARE
                                                GROUP BY cod_client
                                                HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                                        FROM ( SELECT COUNT(*) AS nr_rez
                                                                FROM REZERVARE
                                                                GROUP BY cod_client
                                                        )
                                                )
                                            )
                        )
)
) C.NUME|||C.PRENUME
1 Diaconu Laura
2 Vasilescu Andrei
```

Query Result

All Rows Fetched: 2 in 0 seconds

C.NUME C.PRENUME
1 Diaconu Laura
2 Vasilescu Andrei

Line 10 Column 43 | Insert | Modified | Windows: C

26°C Sunny

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page project

Worksheet Query Builder

```
-- Sa se afiseze numele clientilor care au facut rezervari la toate hotelurile la care s-a cazat clientul
-- cu cele mai multe rezervari facute, incluzandu-l si pe clientul respectiv

-- division

SELECT c.nume || ' ' || c.prenume
FROM REZERVARE r JOIN CLIENT c ON (r.cod_client = c.cod_client)
JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
WHERE h.cod_hotel IN ( SELECT h1.cod_hotel
                        FROM REZERVARE r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                        JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                        JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                        WHERE c1.cod_client = ( SELECT cod_client
                                                FROM REZERVARE
                                                GROUP BY cod_client
                                                HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                                        FROM ( SELECT COUNT(*) AS nr_rez
                                                                FROM REZERVARE
                                                                GROUP BY cod_client
                                                        )
                                                )
                                            )
                        )
)
GROUP BY c.nume, c.prenume
HAVING COUNT(DISTINCT h.cod_hotel) = ( SELECT COUNT(DISTINCT h1.cod_hotel)
                                         FROM REZERVARE r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                                         JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                                         JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                                         WHERE c1.cod_client = ( SELECT cod_client
```

Line 10 Column 43 | Insert | Modified | Windows: C

26°C Sunny

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page project

Worksheet Query Builder

```
-- Sa se afiseze numele clientilor care au facut rezervari la toate hotelurile la care s-a cazat clientul
-- cu cele mai multe rezervari facute, incluzandu-l si pe clientul respectiv

-- division

SELECT c.nume || ' ' || c.prenume
FROM REZERVARE r JOIN CLIENT c ON (r.cod_client = c.cod_client)
JOIN CAMERA cam ON (r.cod_camera = cam.cod_camera)
JOIN HOTEL h ON (cam.cod_hotel = h.cod_hotel)
WHERE h.cod_hotel IN ( SELECT h1.cod_hotel
                        FROM REZERVARE r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                        JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                        JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                        WHERE c1.cod_client = ( SELECT cod_client
                                                FROM REZERVARE
                                                GROUP BY cod_client
                                                HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                                        FROM ( SELECT COUNT(*) AS nr_rez
                                                                FROM REZERVARE
                                                                GROUP BY cod_client
                                                        )
                                                )
                                            )
                        )
)
GROUP BY c.nume, c.prenume
HAVING COUNT(DISTINCT h.cod_hotel) = ( SELECT COUNT(DISTINCT h1.cod_hotel)
                                         FROM REZERVARE r1 JOIN CLIENT c1 ON (r1.cod_client = c1.cod_client)
                                         JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                                         JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                                         WHERE c1.cod_client = ( SELECT cod_client
```

Line 10 Column 43 | Insert | Modified | Windows: C

26°C Sunny

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes 'Oracle Connections' and a selected project named 'grupaISI_011g'. The 'Reports' section is also visible. The main area is the 'Worksheet' tab, titled 'Query Builder', containing a complex SQL query. The query involves multiple joins between tables 'REZERVARE', 'CLIENT', 'CAMERA', and 'HOTEL', along with subqueries and group by clauses. The status bar at the bottom right shows the date and time as '12:02 PM 6/2/2023'.

```

FROM REZERVARE r1 JOIN CLIENT cl ON (r1.cod_client = cl.cod_client)
JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
WHERE cl.cod_client = ( SELECT cod_client
                        FROM REZERVARE
                        GROUP BY cod_client
                        HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                             FROM ( SELECT COUNT(*) AS nr_rez
                                                    FROM REZERVARE
                                                    GROUP BY cod_client
                                             )
                                         )
                        )
GROUP BY c.nume, c.prenume
HAVING COUNT(DISTINCT h.cod_hotel) = ( SELECT COUNT(DISTINCT h1.cod_hotel)
                                         FROM REZERVARE r1 JOIN CLIENT cl ON (r1.cod_client = cl.cod_client)
                                         JOIN CAMERA cam1 ON (r1.cod_camera = cam1.cod_camera)
                                         JOIN HOTEL h1 ON (cam1.cod_hotel = h1.cod_hotel)
                                         WHERE cl.cod_client = ( SELECT cod_client
                                                    FROM REZERVARE
                                                    GROUP BY cod_client
                                                    HAVING COUNT(*) = ( SELECT MAX(nr_rez)
                                                               FROM ( SELECT COUNT(*) AS nr_rez
                                                               FROM REZERVARE
                                                               GROUP BY cod_client
                                                               )
                                         )
                                         )
                                         );

```

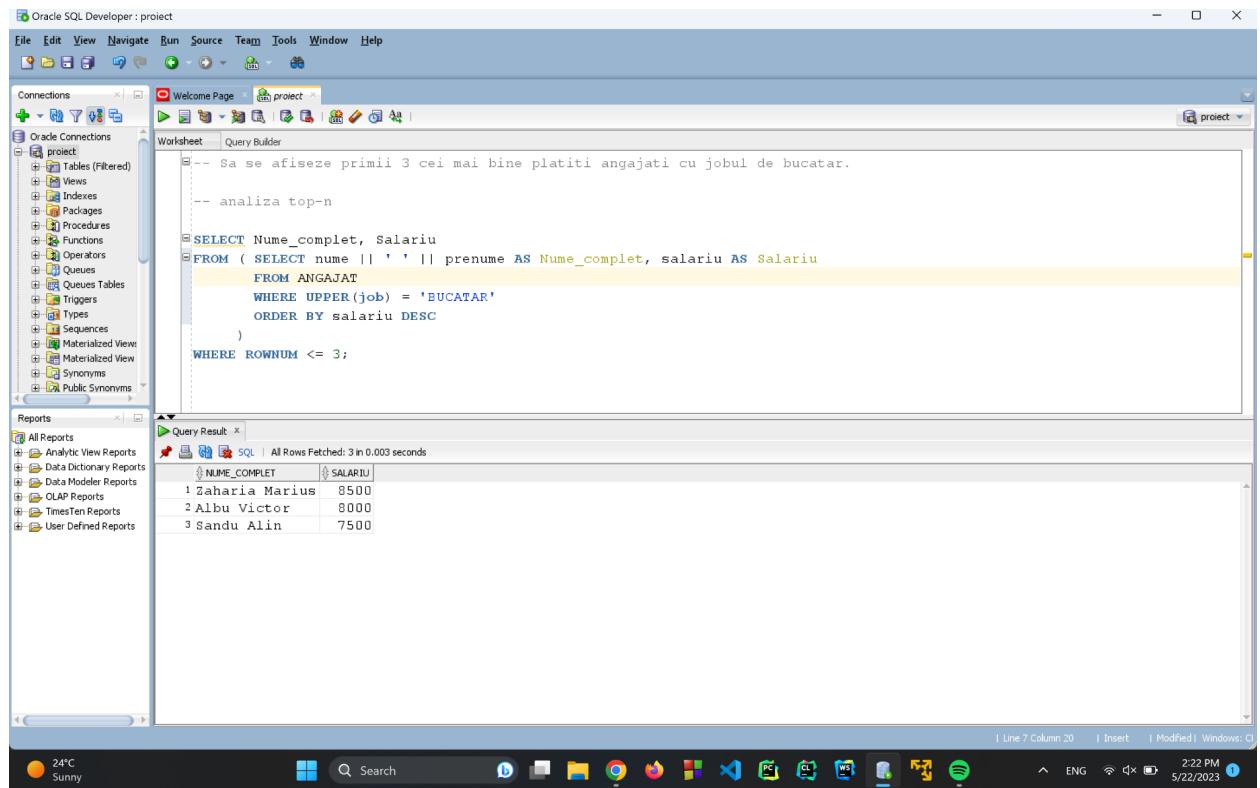
-- Sa se afiseze primii 3 cei mai bine platiti angajati cu jobul de bucatar.

-- analiza top-n

```

SELECT Nume_complet, Salariu
FROM ( SELECT nume || ' ' || prenume AS Nume_complet, salariu AS Salariu
       FROM ANGAJAT
      WHERE UPPER(job) = 'BUCATAR'
        ORDER BY salariu DESC
      )
WHERE ROWNUM <= 3;

```



16. Optimizarea unei cereri, aplicand regulile de optimizare ce deriva din proprietatile operatorilor algebrei relationale. Cererea va fi exprimata prin expresie algebraica, arbore algebraic si limbaj (SQL), atat anterior cat si ulterior optimizarii.

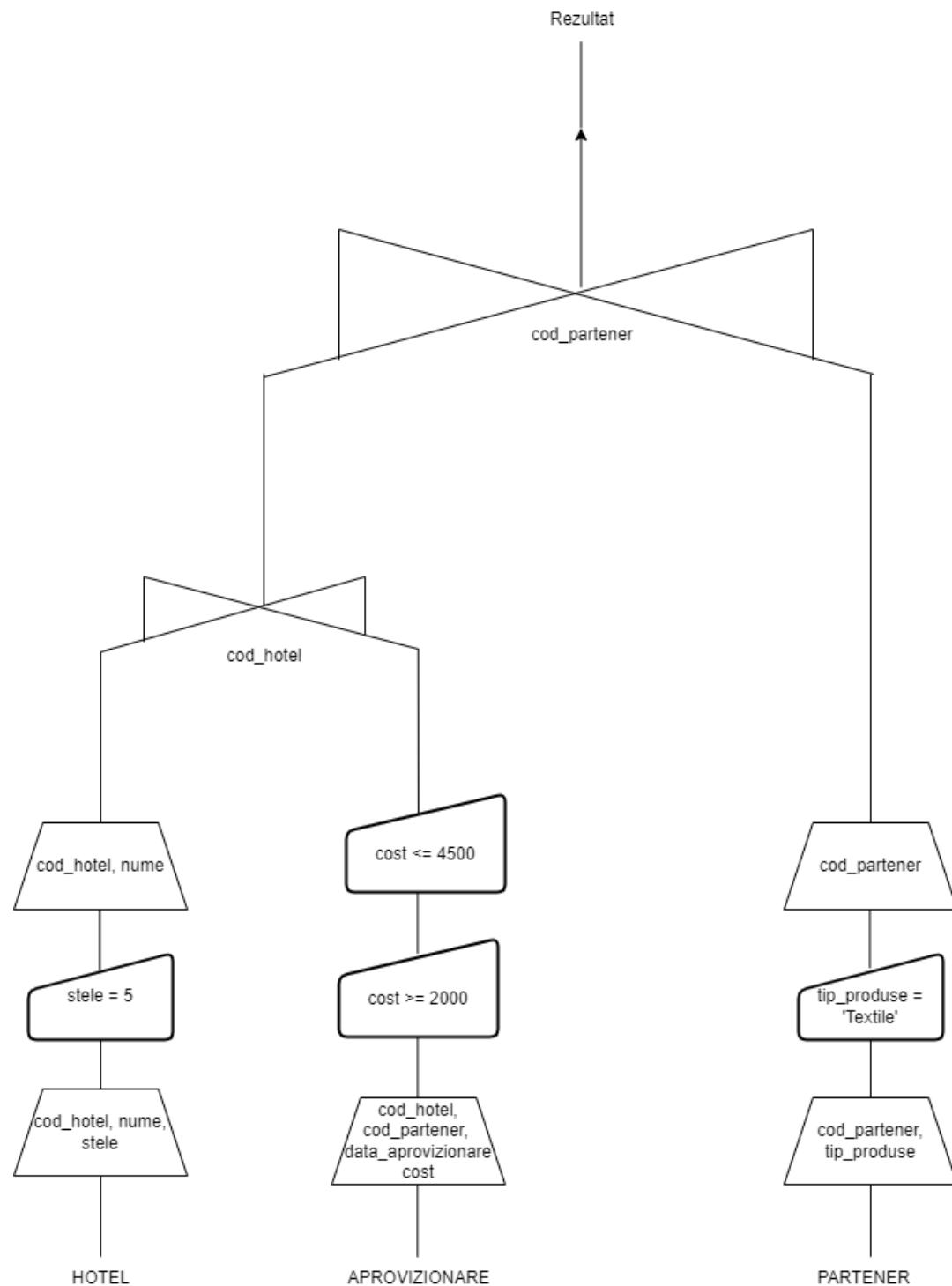
- Sa se afiseze codul si numele hotelurilor, impreuna cu codul partenerului, data aprovizionarii si costul acesteia,
- pentru hotelurile de 5 stele, in care aprovizionarea a fost facuta cu produse textile, iar costul acesteia
- a fost intre 2000 si 4500 de lei.

```
SELECT h.cod_hotel, h.nume, p.cod_partener, a.data_aprovizionare, a.cost
FROM HOTEL h JOIN APROVIZIONARE a ON (h.cod_hotel = a.cod_hotel)
JOIN PARTENER p ON (a.cod_partener = p.cod_partener)
WHERE stele = 5 AND p.tip_produse = 'Textile' AND cost >= 2000 AND cost <= 4500;
```

Expresia algebraica - neoptimizata:

```
R1 = PROJECT(APROVIZIONARE, cod_hotel, cod_partener, data_aprovizionare, cost)
R2 = SELECT(R1, cost >= 2000)
R3 = SELECT(R2, cost <= 4500)
R4 = PROJECT(HOTEL, cod_hotel, nume, stele)
R5 = SELECT(R4, stele = 5)
R6 = PROJECT(R5, cod_hotel, nume)
R7 = JOIN(R3, R6, cod_hotel)
R8 = PROJECT(PARTENER, cod_partener, tip_produse)
R9 = SELECT(R8, tip_produse = 'Textile')
R10 = PROJECT(R9, cod_partener)
Rezultat = JOIN(R7, R10, cod_partener)
```

Arbore algebric - neoptimizat:



Expresie algebraica - optimizata:

```
R1 = SELECT(APROVIZIONARE, cost >= 2000 AND cost <= 4500)
R2 = PROJECT(R1, cod_hotel, cod_partener, data_aprovizionare, cost)
R3 = SELECT(HOTEL, stele = 5)
R4 = PROJECT(R3, cod_hotel, nume)
R5 = JOIN(R2, R4, cod_hotel)
R6 = SELECT(PARTENER, tip_produse = 'Textile')
R7 = PROJECT(R6, cod_partener)
Rezultat = JOIN(R5, R7, cod_partener)
```

Pentru optimizare, am aplicat, mai întâi, Proprietatea 4 (compunerea selecțiilor), pentru a grupa condițiile:

```
R1 = PROJECT(APROVIZIONARE, cod_hotel, cod_partener, data_aprovizionare, cost)
R2 = SELECT(R1, cost >= 2000 AND cost <= 4500)
```

Apoi, folosind Proprietatea 5 (comutarea selecției cu proiecția), ajungem la:

```
R1 = SELECT(APROVIZIONARE, cost >= 2000 AND cost <= 4500)
R2 = PROJECT(R1, cod_hotel, cod_partener, data_aprovizionare, cost)
```

Procedând analog pentru tabelul HOTEL, putem folosi Proprietatea 5 pentru R4, R5 din expresia algebraică anterioară și, apoi, cu Proprietatea 3 (compunerea proiecțiilor), putem restrânge cele 2 proiecții în una singură. Obținem:

```
R3 = SELECT(HOTEL, stele = 5)
R4 = PROJECT(R3, cod_hotel, nume)
```

În mod similar, pentru tabelul PARTENER, folosind Proprietățile 3 și 5, ajungem de la R8, R9, R10 din expresia algebraică neoptimizată la R6, R7 din expresia algebraică optimizată.

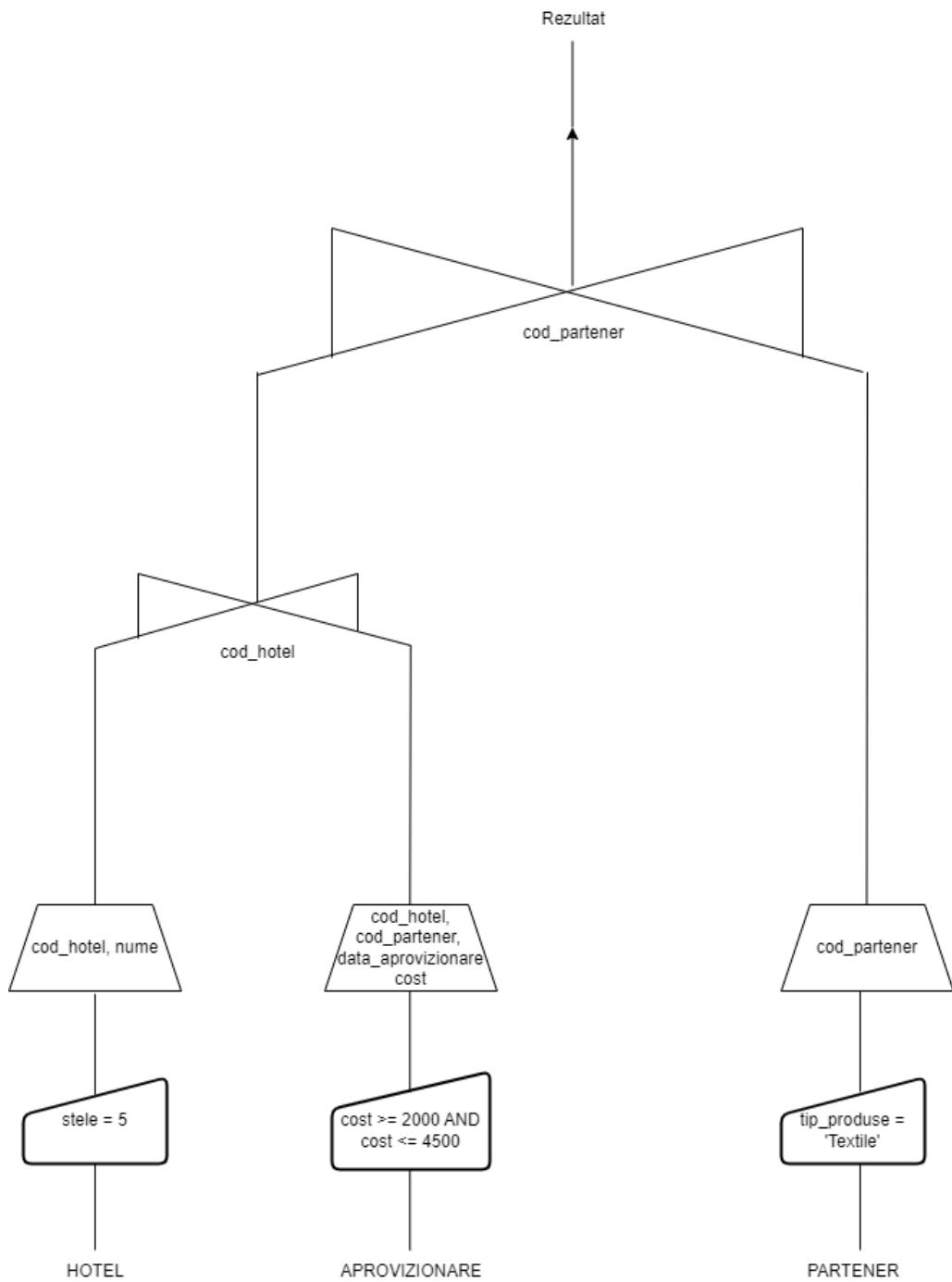
Operația de JOIN intermediară va rămâne la fel, deoarece cele două operații de JOIN pe care le facem sunt la fel de restrictive. Astfel, avem:

```
R5 = JOIN(R2, R4, cod_hotel)
```

În final, ajungem la:

```
Rezultat = JOIN(R5, R7, cod_partener)
```

Arbore algebric - optimizat:



17. a. Realizarea normalizării BCNF, FN4, FN5.

b. Aplicarea denormalizării, justificand necesitatea acesteia.

a. Pentru BCNF:

Toate relațiile se află în BCNF, întrucât fiecare determinant este o cheie candidat și nu există dependențe astfel încât un atribut care nu face parte din cheia primară a relației să determine un alt atribut (care să facă parte din cheia primară compusă).

Presupunând că nu ar putea exista doi parteneri cu același nume care să aibă coduri diferite, un exemplu în care BCNF nu ar mai fi fost respectată ar fi fost dacă, în tabelul APROVIZIONARE, am fi reținut încă un atribut: *nume_partener*. Atunci, acest atribut ar fi determinat *cod_partener*, ce face parte din cheia primară compusă a tabelului, ceea ce nu mai păstrează BCNF.

APROVIZIONARE:

cod_partener#	cod_hotel#	nume_partener	data_aprovizionare#	cost
11	10	Cif	23-APR-2023	1000
14	6	Mobexpert	01-MAR-2022	5000
15	7	Techfit	12-JAN-2023	800

În această situație, pentru a asigura BCNF, ar trebui să separăm tabelul în alte 2 tabele: APROVIZIONARE_1(cod_partener, nume_partener#) și APROVIZIONARE_2(nume_partener#, cod_hotel#, data_aprovizionare#, cost).

Observăm că, în cazul nostru, facem acest lucru prin omiterea acelui atribut și prin plasarea sa în tabelul PARTENER, ce reține informația din APROVIZIONARE_1. Cele două tabele descrise anterior ar arăta astfel:

APROVIZIONARE_1:

cod_partener	nume_partener#
11	Cif
14	Mobexpert
15	Techfit

APROVIZIONARE_2:

cod_hotel#	nume_partener#	data_aprovizionare#	cost
10	Cif	23-APR-2023	1000
6	Mobexpert	01-MAR-2022	5000
7	Techfit	12-JAN-2023	800

Pentru FN4:

O relație se află în FN4 dacă:

- 1) se află în BCNF (ceea ce am stabilit anterior)
- 2) nu conține relații m:n independente

De fapt, relațiile care trebuie verificate sunt cele care definesc relații many-to-many între tabele (adică tabelele asociative), observând că celelalte tabele se află în FN4. Tabelul OFERTA conține doar cod_tip și cod_facilitate, așa că nu conține relații m:n independente.

Pentru tabelul APROVIZIONARE, cheia primară este compusă din trei chei (mai există atributul cost, dar acesta nu pune nicio problemă). Totuși, nu putem spune că avem multidependență cod_partener $\rightarrow\rightarrow$ data_aprovizionare, întrucât partenerii nu au niște date stabilite la care să aprovizioneze, ci o fac după cererea hotelului respectiv. Astfel, tabelul este în FN4.

Ultimul tabel de analizat este REZERVARE, în care observăm că există dependențe multiple:

cod_client $\rightarrow\rightarrow$ cod_camera
 cod_client $\rightarrow\rightarrow$ cod_pachet

Cealaltă componentă a cheii primare compuse (data_sosire), precum și atributul rămas (pret) nu influențează aspectul de FN4 al relației.

Pentru a aduce relația în FN4, vom descompune relația în 2 relații care se află în FN4:

REZERVARE = JOIN(REZERVARE_1, REZERVARE_2), unde:
 REZERVARE_1(cod_client#, cod_camera#, data_sosire#, pret) și
 REZERVARE_2(cod_client#, cod_pachet#)

Practic, descompunerea arată astfel:

REZERVARE:

cod_client#	cod_camera#	cod_pachet#	data_sosire#	pret
20	110	150	05-JUN-2020	1800
20	110	153	09-AUG-2020	2300
20	111	150	23-NOV-2019	2700
20	111	153	08-FEB-2021	4500

REZERVARE_1:

cod_client#	cod_camera#	data_sosire#	pret
20	110	05-JUN-2020	1800
20	110	09-AUG-2020	2300
20	111	23-NOV-2019	2700
20	111	08-FEB-2021	4500

REZERVARE_2:

cod_client#	cod_pachet#
20	150
20	153

Pentru FN5:

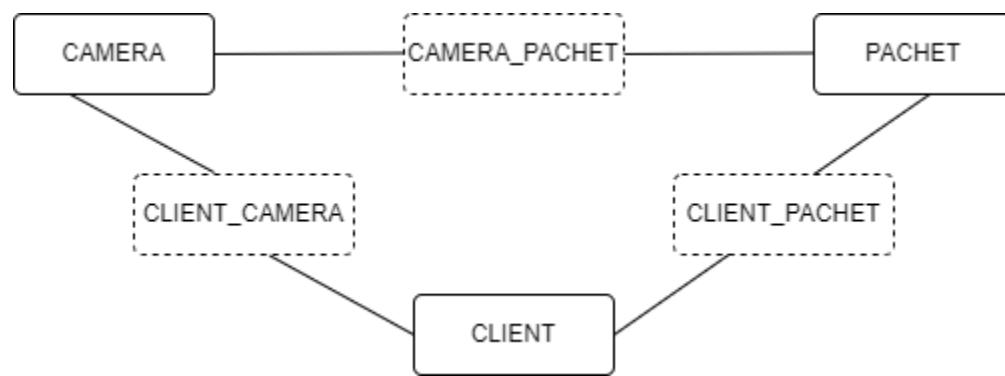
O relație se află în FN5 dacă:

- 1) se află în FN4 (ceea ce am discutat anterior)
- 2) nu conține dependențe ciclice

Mai exact, tabelele ce trebuie verificate sunt acelea care ar putea înlocui o relație de tip 3 prin mai multe relații de tip 2, celelalte aflându-se în FN5 din lipsa posibilelor dependențe ciclice.

În diagrama noastră, există o singură situație în care apare ideea de relație de tip 3, prin tabelele CLIENT, CAMERA, PACHET. Implementând o astfel de relație printr-un singur tabel asociativ ce descrie relația de tip 3 (tabelul REZERVARE, transformat în REZERVARE_1 și REZERVARE_2 pentru FN4), putem spune că toate relațiile se află în FN5.

Un scenariu în care FN5 nu ar mai fi fost respectată, ar fi fost dacă, în loc de tabelul REZERVARE, am fi gestionat legăturile dintre cele 3 tabele prin 3 tabele asociative, ce descriu relațiile many-to-many dintre fiecare 2 tabele:



Practic, informația din tabelul REZERVARE ar fi fost împărțită în 3 tabele, având: $REZERVARE = \text{JOIN}(\text{CLIENT_PACHET}, \text{CLIENT_CAMERA}, \text{CAMERA_PACHET})$.

Totuși, o astfel de implementare ar fi generat o dependență ciclică, ceea ce ar fi făcut ca diagrama să nu se mai afle în FN5.

b. Necesitatea denormalizării:

Deși denormalizarea poate mări redundanța, este important beneficiul major pe care îl aduce: reducerea numărului de join-uri făcute pentru rezolvarea unei interogări, ce duce la micșorarea timpului de execuție.

De asemenea, este și mai important să nu pierdem informații în urma normalizărilor.

Spre exemplu, normalizarea făcută pentru a ajunge în FN4 nu era necesară, deoarece pierdem informații despre rezervările făcute. Dacă folosim tabelele descrise anterior, observăm că nu mai putem răspunde la întrebări de tipul: cu ce pachet a făcut clientul rezervarea la o anumită cameră? Chiar și dacă am reține data sosirii în ambele tabele, rămâne posibilitatea ca un client să fi făcut mai multe rezervări pentru mai multe camere în aceeași zi.

Pe lângă acest fapt, presupunând că ar exista o metodă de a păstra toată informația în mod corect, ar fi nevoie de multe operații de join, ce sunt costisitoare, pentru a afla datele pentru fiecare rezervare în parte.

De aceea, denormalizarea, în acest caz, ar constitui în inversarea procedeului de transformare în FN4, păstrând toate tabelele aşa cum le-am descris inițial.