

Proiect Baze de Date

George Radu

1 Introducere - ex1

Detaliile din acest proiect se referă la proiectarea unui model de date ce furnizează informații despre gestiunea companiilor de transporturi rutiere de mărfuri din România.

Modelul de date va gestiona informații legate de transporturile de mărfuri ale unor companii client de un depozit la altul. Există firme de pază care se ocupă de paza depozitelor. O firmă de pază alocă fiecărui depozit o echipă de pază.

Transporturile de marfă sunt realizate de o firmă de transport prin intermediul angajaților profesioniști care pot avea mai multe categorii de permise și se pot opri la mai multe popasuri pentru odihnă/masă/alimentare. Transportul de marfă începe de la un depozit și se termină la alt depozit, în fiecare depozit accesul camionului este permis după o verificare a echipei de pază.

2 Restricții de funcționare - ex2

Modelul de date respectă anumite restricții de funcționare.

- Orice depozit are un singur deținător(o companie client ori o firmă de transport) și poate fi folosit pentru depozitarea de mărfuri.
- O companie client sau o firmă de transport poate deține mai multe depozite.
- Unui depozit îi este alocată o unică locație.
- Un depozit este păzit la un moment dat de o singură echipă de pază.

- Un depozit are o listă de inventar cu ce mărfuri au intrat/ieșit împreună cu data și ora.
- Un lot de marfă poate să fie transportat de mai multe ori, nu neapărat îl același transport, și nu neapărat între aceleași depozite.
- O echipă de pază aparține unei firme de pază, firmă de pază care poate avea mai multe echipe de pază, o firmă putând astfel să păzească mai multe depozite aflate în diferite locații.
- Un angajat poate să fie ori paznic ori șofer.
- Un transport de marfă se realizează prin intermediul unui camion încărcat cu marfă de la un depozit la altul.
- Un șofer poate să oprească în timpul transportului la mai multe popasuri pentru odihnă/mâncare sau alimentare. La popasuri pot veni mai mulți șoferi.
- Un popas se află într-o unică locație și poate fi o benzinărie, un motel sau un restaurant. Nu ne interesează detalii specifice despre un popas cum ar fi ce mâncare vinde un restaurant, sau cât costa o cameră la un motel.
- Un șofer poate să aibă mai multe permise pentru mai multe categorii, poate chiar să i se anuleze/expire un permis și să trebuiască să îl obțină din nou.
- Un șofer trebuie să conducă un camion, dar poate să conducă mai multe camioane(nu în același timp) în funcție de problemele tehnice legate de camion și necesitățile firmei.
- O firmă de transport deține mai multe camioane, iar un camion este deținut de o singură firmă de transport.
- Transportul de marfă se face între 2 depozite, se transportă mai multe loturi de marfă. Pentru un transport vom reține și data din momentul în care a început transportul. Pentru fiecare transport se va folosi un singur camion condus de un șofer profesionist.

3 Entități - ex3

Pentru modelul de date referitor la gestiunea companiilor de transporturi rutiere de mărfuri, structurile: DEPOZIT, LOCATIE, POPAS, FIRMA, CLIENT, PAZA, TRANSPORT, ECHIPA_PAZA, INVENTAR_DEPOZIT, INVENTAR_TRANSPORT, LOT_MARFA, CAMION, ANGAJAT, PAZNIC, SOFER, PERMIS reprezintă entități.

Vom prezenta entitățile modelului de date, realizând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

Toate entitățile care vor fi precizate sunt independente, cu excepția entităților dependente ECHIPA_PAZA, PERMIS și INVENTAR_TRANSPORT și a subentităților SOFER, PAZNIC, CLIENT, PAZA, TRANSPORT.

- DEPOZIT = entitatea descrie o construcție al cărei proprietar este o firmă client sau de transport, în care pot fi depozitate mărfuri. Cheia primară a entității este id_depozit.
- LOCATIE = entitatea descrie informații despre locația fizică a unei construcții, fie ea depozit sau popas. Cheia primară a entității este id_locatie.
- POPAS = reprezintă un loc în care un șofer se poate opri pentru odihnă. Cheia primară a entității este id_popas.
- FIRMA = entitatea descrie o persoană juridică din România, supraentitate pentru CLIENT, PAZA și TRANSPORT. Cheia primară a entității este id_firma.
- CLIENT = persoană juridică beneficiară a unor servicii de transport oferite de o firmă de transport specializată. Cheia primară a entității este id_firma.
- PAZA = persoana juridică ce asigură securitatea depozitelor prin intermediul unor echipe de pază formate din paznici profesioniști. Cheia primară a entității este id_firma_paza.
- TRANSPORT = persoană juridică responsabilă de transportul de mărfuri în siguranță pentru companii client de la un depozit la altul. Pentru transporturi firma pune la dispoziție șoferilor săi mai multe camioane. Cheia primară a entității este id_firma_transport.

- ECHIPA_PAZA = entitatea descrie informații despre o echipă formată din paznici responsabili de securitatea unui depozit. Cheia primară a entității este id echipa_paza.
- INVENTAR_DEPOZIT = entitatea descrie informații despre inventarierea unui depozit, ce loturi au intrat în inventariere, când au intrat și când au ieșit. Cheia primară a entității este id_inventar_depozit.
- INVENTAR_TRANSPORT = entitatea descrie informații despre ce lot de marfă a fost implicat într-un transport între 2 depozite. Cheia primară a entității este cheia compusa dintre id_inventar_transport și id_lot_marfa.
- LOT_MARFA = entitatea descrie informații despre un lot de marfă transportată sau depozitată. Cheia primară a entității este id_marfa.
- CAMION = vehicul de mare tonaj pentru transport marfă. Cheia primară a entității este id_camion.
- ANGAJAT = persoană fizică angajată a unei companii fie de pază fie de transport pentru a-și oferi serviciile în schimbul unei remunerații. Cheia primară a entității este id_angajat.
- PAZNIC = subentitate a entității ANGAJAT, lucrează pentru o companie de pază și este responsabil de securitatea unui depozit din cadrul echipei sale. Cheia primară a entității este id_angajat.
- SOFER = subentitate a entității ANGAJAT, lucrează pentru o firmă de transport, îi este alocat un camion și este responsabil pentru transportul în siguranță a unei remorci cu marfă. Cheia primară a entității este id_angajat.
- PERMIS = entitate ce descrie informații despre ce tipuri de permis are un șofer și când le-a obținut. Cheia primară a entității este una compusă din id_permis + id_angajat.

4 Relații - ex4

Se vor prezenta relațiile modelului de date, dând o descriere completă a fiecăreia. Denumirile acestor legături între entități sunt alese sugestiv. Pentru fiecare relație se va preciza cardinalitatea minimă și maximă.

- DEPOZIT_se_află_LOCATIE = relație ce leagă entitățile DEPOZIT și LOCATIE, reflectând legătura dintre acestea (un depozit se află plasat într-o locație unică din țară). Ținând cont de condițiile impuse modelului un depozit trebuie să se afle într-o singură locație. Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:1.
- POPAS_se_află_LOCATIE = relație ce leagă entitățile POPAS și LOCATIE, reflectând legătura dintre acestea (un popas se află plasat într-o locație unică din țară). Ținând cont de condițiile impuse modelului un popas trebuie să se afle într-o singură locație. Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:1.
- FIRMA_deține_DEPOZIT = relație ce leagă entitățile FIRMA și DEPOZIT, reflectând legătura dintre acestea (un depozit este deținut de o firmă). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:M.
- PAZA_apartține_ECHIPA_PAZA = relație ce leagă entitățile PAZA și ECHIPA_PAZA, reflectând legătura dintre acestea (o echipă de pază aparține unei firme de pază). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:M.
- ECHIPA_PAZA_păzește_DEPOZIT = relație ce leagă entitățile ECHIPA_PAZA și DEPOZIT, reflectând legătura dintre acestea (o echipă de pază trebuie să păzească un depozit pentru a permite accesuri în depozit doar de staff autorizat). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:1 (Conform condițiilor impuse modelului un depozit va fi păzit la un moment în timp doar de o echipă de pază).
- PAZNIC_apartține_ECHIPA_PAZA = relație ce leagă entitățile PAZNIC și ECHIPA_PAZA, reflectând legătura dintre acestea (un paznic trebuie să facă parte dintr-o echipă de pază trimisă la un depozit, paznicul lucrează pentru firmă de pază de care aparține echipa). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă M:1 (un paznic face parte dintr-o singură echipă, iar o echipă este formată din mai mulți paznici).

- INVENTAR_DEPOZIT_apartine_DEPOZIT = relație ce leagă entitățile INVENTAR_DEPOZIT și DEPOZIT, reflectând legătura dintre acestea (un depozit trebuie să țină inventarul la ce loturi de marfă intră și iese și când s-au întâmplat aceste evenimente). Relația are cardinalitatea minimă 0:1 și cardinalitatea maximă M:1 (inițial un depozit abia construit poate să nu conțină niciun record de inventariere, urmând pe parcurs să aibă).
- INVENTAR_DEPOZIT_apartine_LOT_MARFA = relație ce leagă entitățile INVENTAR_DEPOZIT și LOT_MARFA, reflectând legătura dintre acestea (un lot de marfă va fi înregistrat în lista de inventar a unui depozit atunci când va fi adus în depozit fizic). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă M:1 (Același lot de marfă poate să fie transferat între depozite din puncte cheie din țară în funcție de necesitățile firmei client).
- LOT_MARFA_apartine_INVENTAR_TRANSPORT = relație ce leagă entitățile LOT_MARFA și INVENTAR_TRANSPORT, reflectând legătura dintre acestea (pentru fiecare transport de marfă se face un inventar din care fac parte mai multe loturi de marfă, un lot de marfă poate să fie transportat de mai multe ori). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:M (ținând cont de condițiile impuse modelului entitatea INVENTAR_TRANSPORT va conține doar un lot de marfă și din ce lot face parte).
- CAMION_transportă_INVENTAR_MARFĂ_de_la_DEPOZIT_la_DEPOZIT = relație de tip 3 leagă entitățile DEPOZIT, DEPOZIT, INVENTAR_MARFĂ și CAMION, reflectând ce marfă a fost transportată de la un depozit de plecare la unul de sosire, transportul fiind făcut cu un camion. Denumirea acestei relații va fi transport.
- TRANSPORT_detine_CAMION = relație ce leagă entitățile TRANSPORT și CAMION, reflectând legătura dintre acestea (o firmă de transport trebuie să dețină camioane). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:M.
- SOFER_conduce_CAMION = relație ce leagă entitățile SOFER și CAMION, reflectând legătura dintre acestea (un șofer de la o firmă de transport trebuie să conducă cel puțin un camion pentru a-și îndeplini sarcinile de la muncă). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă M:M.
- ANGAJAT_lucraza_la_FIRMA = relație ce leagă entitățile ANGAJAT și FIRMA, reflectând legătura dintre acestea (un angajat/o persoană

fizică este angajat/ă la o firmă, acesta își oferă serviciile în schimbul unei remunerații). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă M:1.

- SOFER_detine_PERMIS = relație ce leagă entitățile SOFER și PERMIS, reflectând legătura dintre acestea (un șofer pentru a se putea angaja are nevoie de un permis care să îi ofere dreptul de a conduce un camion, șoferul poate să aibă mai multe categorii de permise, în funcție de necesitățile sale). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:M.
- SOFER_istoric_popasuri_POPAS = relație ce leagă entitățile SOFER și POPAS, reflectând legătura dintre acestea (un șofer nu poate să conducă mai mult de 8h conform legii, motiv pentru care trebuie să mai ia pauze de odihnă, pauze pe care le poate lua la un motel sau în parcarearea unui restaurant sau benzinării de unde poate să și mănânce și să se întâlnească cu alți șoferi). Relația are cardinalitatea minimă 0:0 și cardinalitatea maximă M:M.

5 Atribute - ex5

- Entitatea independentă LOCATIE are ca atribute:
 - id_locatie = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unei locații, reprezintă cheia primară.
 - județ = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele județului.
 - localitate = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele localității.
 - stradă = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele străzii.
 - nr = variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul locației.
- Entitatea independentă DEPOZIT are ca atribute:
 - id_depozit = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui depozit, reprezintă cheia primară.
 - id_locatie = variabilă de tip întreg, de lungime maximă 10, care

reprezintă id-ul locației pentru depozit. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul LOCATIE.

- id_firma = variabilă de tip întreg, de lungime maximă 12, care reprezintă id-ul firmei care deține depozitul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.
- id echipa_paza = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul echipei de pază al depozitului. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ECHIPA_PAZA.

- Entitatea independentă POPAS are ca atribute:

- id_popas = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui popas, reprezintă cheia primară.
- id_locatie = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul locației pentru popas. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul LOCATIE.
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele popasului.
- tip_popas = variabilă de tip caracter, de lungime maximă 10. Atributul poate lua valorile: BENZINĂRIE, MOTEL, RESTAURANT; nu poate fi *null*.

- Entitatea independentă FIRMA are ca atribute:

- id_firma = variabilă de tip întreg, de lungime maximă 12, care reprezintă id-ul firmei, reprezintă cheia primară.
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele firmei, numele unei firme este unic.
- tip_firmă = variabilă de tip caracter, de lungime maximă 10, care poate lua valorile: CLIENT, PAZA, TRANSPORT; nu poate fi *null*.

- Entitatea ECHIPA_PAZA are ca atribute:

- id echipa_paza = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei echipe din cadrul firmei de pază, reprezintă cheia primară.
- id_firma = variabilă de tip întreg, de lungime maximă 12, care reprezintă id-ul firmei de pază. Atributul trebuie să corespundă

la o valoare a cheii primare din tabelul PAZA.

- Entitatea INVENTAR_DEPOZIT are ca attribute:
 - id_inventar_depozit = variabilă de tip întreg de lungime maximă 10, care reprezintă id-ul unei intrări de marfă în inventar, reprezintă cheia primară.
 - id_depozit = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul depozitului în care se află marfa. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul .
 - id_lot_marfa = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul lotului de marfă care a intrat în inventar. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul LOT_MARFA.
 - data_sosire = variabilă de tip data calendaristică, care reprezintă data la care a intrat marfa în inventariere; are valoarea default data în care se adaugă un nou element, i.e. *sysdate*.
 - ora_sosire = variabilă de tip ora, care reprezintă ora la care a intrat marfa în inventarierea; are valoarea default ora la care s-a adăugat un nou element.
 - data_plecare = variabilă de tip data calendaristică, care reprezintă data la care a ieșit marfa din inventariere.
 - ora_plecare = variabilă de tip ora, care reprezintă ora la care a ieșit marfa din inventariere.
- Entitatea LOT_MARFA are ca attribute:
 - id_lot_marfa = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui lot de marfă, reprezintă cheia primară.
 - nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele mărfii din lot; nu poate fi *null*.
 - cantitate = variabilă de tip real, de lungime maximă 10, care reprezintă cantitatea lotului de marfă, nu poate fi *null*.
- Entitatea INVENTAR_TRANSPORT are ca attribute:
 - id_inventar_transport = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unei intrări de inventar al unui transport; face parte din cheia primară compusă.

- id_lot_marfa = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui lot de marfă care face parte din inventarul unui transport. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul LOT_MARFA, face parte din cheia primară compusă.
- id_transport = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul transportului din care face parte această listare. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TRANSPORT.
- Entitatea TRANSPORT are ca atribute:
 - id_transport = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui transport de marfă între 2 depozite, reprezintă cheia primară.
 - depozit_plecure = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul depozitului de unde începe transportul și este preluată marfa. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul DEPOZIT.
 - depozit_destinație = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul depozit unde este livrată marfa. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul DEPOZIT.
 - id_camion = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul camionului folosit pentru transport. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CAMION.
 - data_plecure = variabilă de tip data calendaristică, care reprezintă data la care a plecat camionul din depozitul de plecare; are valoarea default data în care s-a adăugat un nou element, i.e. *sysdate*.
 - ora_plecure = variabilă de tip ora, care reprezintă ora la care a plecat camionul din depozitul de plecare; are valoarea default ora la care s-a adăugat un nou transport.
- Entitatea CAMION are ca atribute:
 - id_camion = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui camion, reprezintă cheia primară.
 - id_firma = variabilă de tip întreg, de lungime maximă 12, care

reprezintă id-ul firmei de transport care deține camionul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TRANSPORT.

- marca = variabila de tip caracter, de lungime maxima 10, care reprezintă firma care a produs camionul.
- nr_inmatriculare = variabila de tip caracter, de lungime maxima 10, reprezintă un identificator unic pentru autovehicule. Acest atribut poate fi schimbat (de obicei cand se schimba detinatorul) motiv pentru care nu reprezintă parte din cheia primară.

- Entitatea ANGAJAT are ca attribute:

- id_angajat = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui angajat, reprezintă cheia primară.
- nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele unui angajat.
- prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele unui angajat.
- nr_telefon = variabilă de tip caracter, de lungime maximă 14, care reprezintă numărul de telefon al unui angajat.
- data_angajare = variabilă de tip data calendaristică, care reprezintă data în care a fost angajat o persoană; are valoarea default data în care a fost adăugat un angajat în baza de date, i.e. *sysdate*.
- salariu = variabilă de tip real, de lungime maximă 10, care reprezintă salariul angajatului.
- id_firma = variabile de tip întreg, de lungime maximă 12, care reprezintă id-ul firmei la care lucrează angajatul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.
- tip_angajat = variabilă de tip caracter, de lungime maximă 6, care reprezintă tipul angajatului. Poate să fie PAZNIC sau SOFER.

- Entitatea PAZNIC are aceleași attribute ca supraentitatea din care face parte + atributul id echipa_paza (= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul echipei de pază din care face parte paznicul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ECHIPA_PAZA).

- Entitatea SOFER are aceleași atribute ca supraentitatea din care face parte. Nu are alte atribute adiționale.
- Entitatea ISTORIC_CAMIOANE_CONDUSE are ca atribute:
 - id_istoric_camioane_conduse = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unei intrări în istoric, reprezintă cheia primară.
 - id_angajat = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui șofer. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SOFER.
 - id_camion = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul camionului condus. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CAMION; are valoarea default data în care s-a adăugat un element, i.e. *sysdate*.
 - data_inceput = variabilă de tip data calendaristică, care reprezintă data la care a început un șofer să conducă un camion al firmei de transport.
 - data_sfarsit = variabilă de tip data calendaristică, care reprezintă data la care un șofer nu mai conduce un camion.
- Entitatea PERMIS are ca atribute:
 - id_permis = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui permis; face parte din cheia primară compusă.
 - id_angajat = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui angajat care deține un permis. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SOFER, face parte din cheia primară compusă.
 - categorie = variabilă de tip caracter, de lungime maximă 3, care reprezintă tipul de permis care poate fi deținut de o persoană. Atributul poate lua valorile AM, A1, A2, A, B1, B, BE, C1, C1E, C, CE, D1, D1E, D, DE, Tr, Tb sau Tv.
 - data = variabilă de tip data calendaristică, care reprezintă data la care a fost emis permisul pentru o persoană; are ca valoare default data în care a fost introdus un element în baza de date, i.e. *sysdate*.

- ora = variabilă de tip ora, care reprezintă ora la care a fost emis permisul pentru o persoană; are ca valoare default ora la care a fost introdus un element în baza de date.
- Entitatea ISTORIC_POPASURI are ca attribute:
 - id_istoric_popas = variabilă de tip întreg, de lungime maximă 20, care reprezintă id-ul unui istoric în popas; face parte din cheia primară compusă.
 - id_popas = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui popas la care s-a oprit un șofer. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul POPAS; face parte din cheia primară compusă.
 - id_angajat = variabilă de tip întreg, de lungime maximă 10, care reprezintă id-ul unui șofer care s-a oprit la un popas. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SOFER; face parte din cheia primară compusă.
 - data_sosire = variabilă de tip data calendaristică, care reprezintă data la care a ajuns un șofer la un popas; are ca valoare default data în care a fost introdus un element, i.e. *sysdate*; face parte din cheia primară compusă.
 - ora_sosire = variabilă de tip ora, care reprezintă ora la care a ajuns un șofer la un popas; are ca valoare default ora la care a fost introdus un element; face parte din cheia primară compusă.
 - data_plecare = variabilă de tip data calendaristică, care reprezintă data la care a plecat un șofer de la un popas; are ca valoare default data la care a fost introdus un element, se consideră ca majoritatea opririlor sunt de durată scurtă/medie și cazurile în care o oprire se lungeste pe ≥ 2 zile fiind rare.
 - ora_plecare = variabilă de tip ora, care reprezintă ora la care a plecat un șofer de la un popas; analog data_plecare.

Mai sunt de prezentat următoarele relații m:m care se vor transforma în tabele asociative în implementare:

- Relația CAMION_transportă_INVENTAR_MARFĂ_de_la_DEPOZIT_la_DEPOZIT are ca attribute: id_transport, depozit_plecare, depozit_destinație, camion, data_plecare, ora_plecare.
- Relația SOFER_conduce_CAMION are ca attribute:

id_istoric_camioane_conduse, id_angajat, id_camion, data_inceput,
data_sfarsit.

- Relația SOFER_istoric_popasuri_POPAS are ca atribut:
- id_istoric_popas, id_popas, id_angajat, data_sosire, ora_sosire,
data_plecure, ora_plecure.

6 Diagrama entitate-relație E/R - ex6

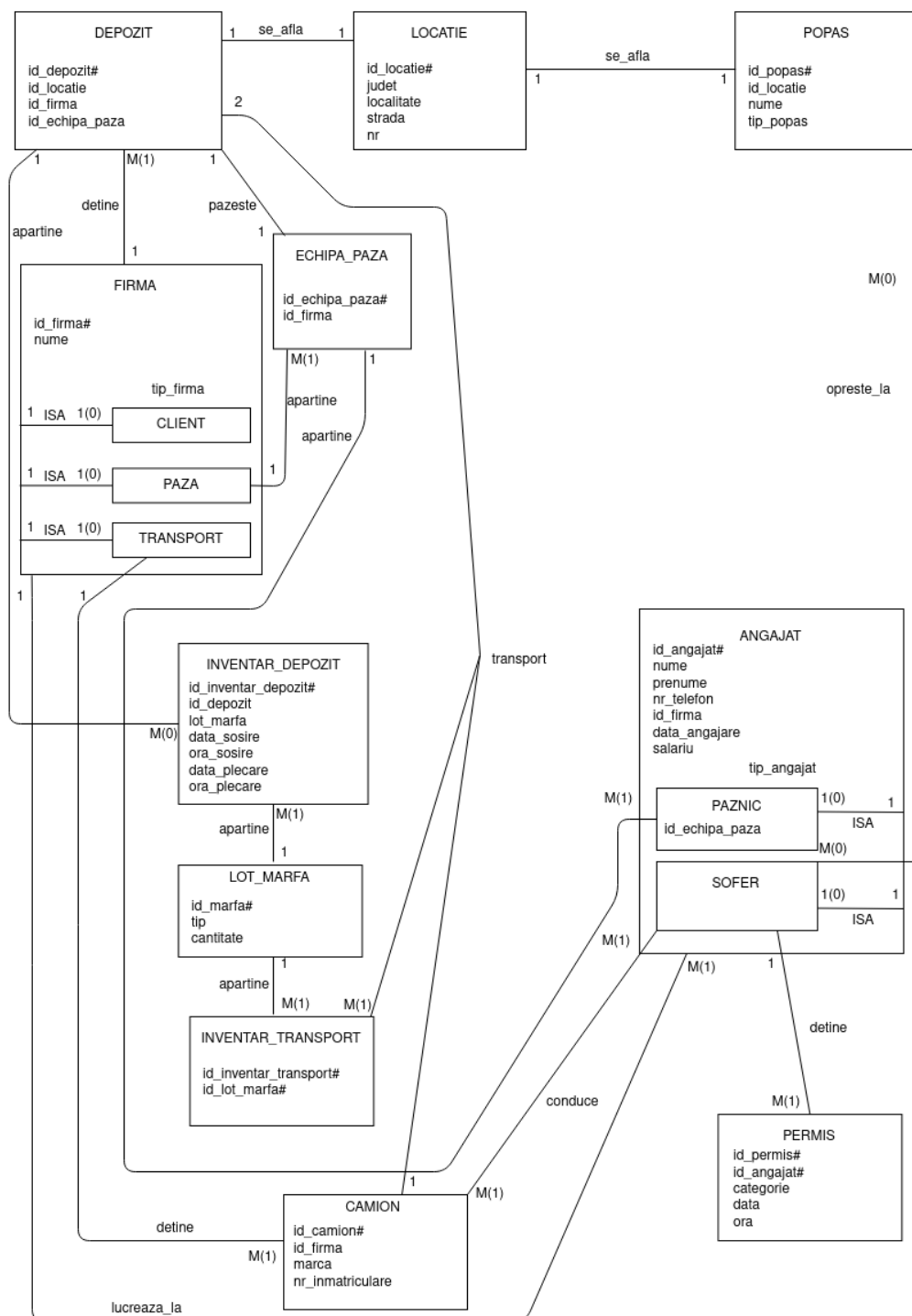


Figura 1

7 Diagrama conceptuală - ex7

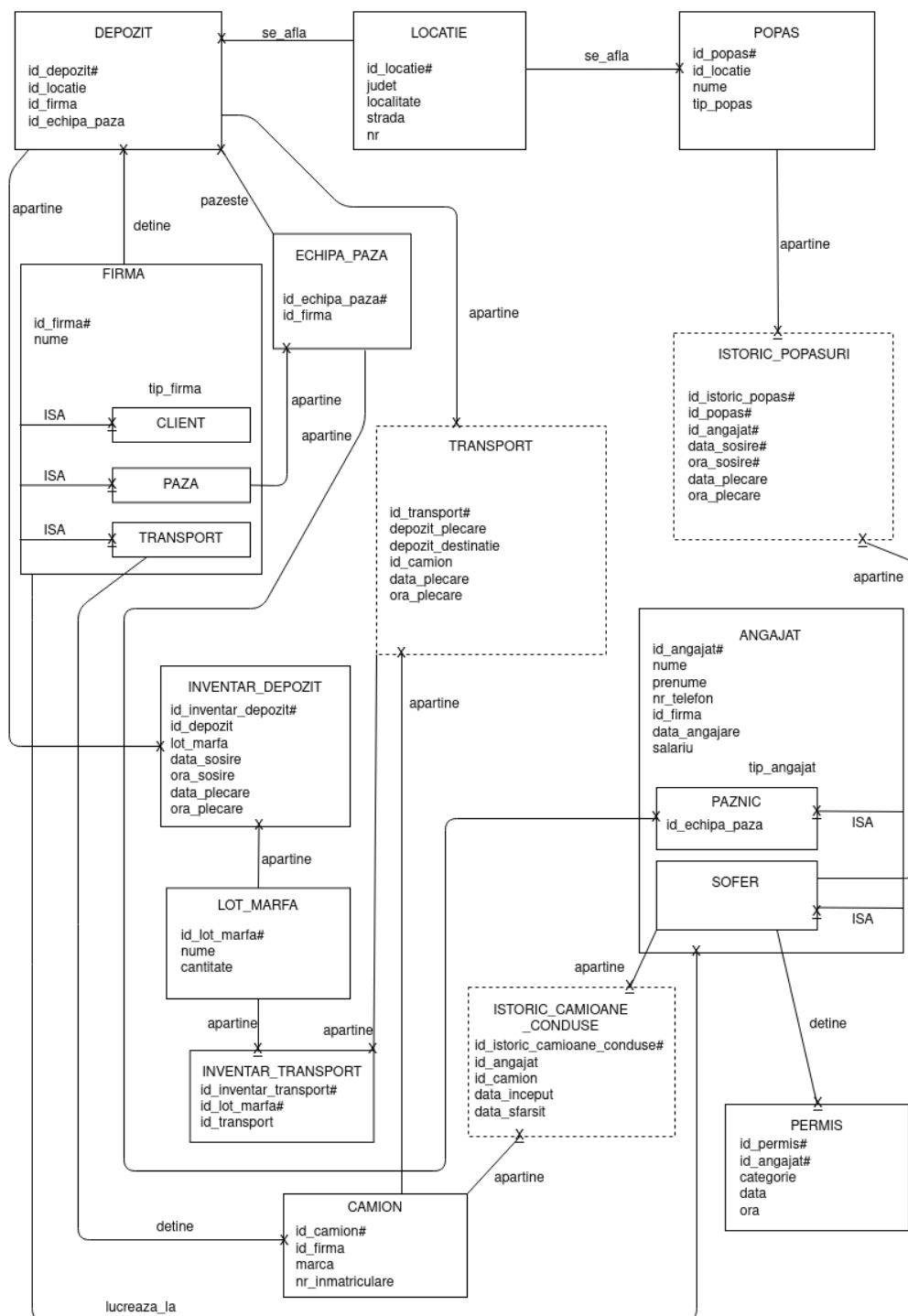


Figura 2

8 Schemele relaționale - ex8

Schemele relaționale corespunzătoare diagramei conceptuale din Figura 2 sunt următoarele:

- DEPOZIT(id_depozit#, id_locatie, id echipa_paza, id_firma)
- LOCATIE(id_locatie#, județ, localitate, stradă, nr)
- POPAS(id_popas#, id_locatie, nume, tip_popas)
- FIRMA(id_firma#, nume, tip_firmă)
- CLIENT(id_firma#)
- PAZA(id_firma#)
- TRANSPORT(id_firma#)
- ECHIPA_PAZA(id echipa_paza#, id_firma_paza)
- INVENTAR_DEPOZIT(id_inventar_depozit#, id_depozit, id_lot_marfa, data_sosire, ora_sosire, data_plecure, ora_plecure)
- INVENTAR_TRANSPORT(id_inventar_transport#, id_marfa#, id_transport)
- LOT_MARFA(id_lot_marfa#, nume, cantitate)
- CAMION(id_camion#, id_firma, marca, nr_inmatriculare)
- TRANSPORT(id_transport#, depozit_plecure, depozit_destinație, id_camion, data_plecure, ora_plecure)
- ANGAJAT(id_angajat#, nume, prenume, nr_telefon, data_angajare, salariu, id_firma, tip_angajat)
- PAZNIC(id_angajat#, id echipa_paza)
- SOFER(id_angajat#)
- PERMIS(id_permis#, id_angajat#, categorie, data, ora)
- ISTORIC_CAMIOANE_CONDUSE(id_istoric_camioane_conduse#, id_angajat, id_camion, data_inceput, data_sfarsit)
- ISTORIC_POPASURI(id_istoric_popas#, id_popas#, id_angajat#, data_sosire#, ora_sosire#, data_plecure, ora_plecure)

9 Forme normale - ex 9 + 18

Forma normala 1 - exemplu 1

Se dă relația *SOFER_detine_PERMIS* în care un șofer poate deține mai multe permise. S-a plecat de relația de tipul (obs. nu sunt trecute toate atributele unui angajat, doar câteva pentru a nu mări tabelul și a face toate datele cât mai lizibile și clare):

id_angajat#	nume	prenume	..	tip_angajat	id_permis#	categorie	data	ora
20000	Ion	Vasile	..	SOFER	1	C, B, A	06-06-2012, 06-06-2013, 06-06-2017	12:00, 12:00, 12:00
20020	Ovidie	Ion	..	SOFER	2	C	06-06-2018	12:00
20100	Constantin	Gheorghe	..	SOFER	3	B, C	06-06-2010, 06-06-2012	12:00, 12:00

Table 1: Relația nu e în FN1

în care se observă că atributelor *categorie*, *data* și *ora* le corespund o listă de valori, i.e. nu le corespund o valoare indivizibilă (atomică), de unde se trage concluzia că relația *SOFER_detine_PERMIS* nu se află în forma normala 1 (FN1). Se aduce relația în FN1 spărgând atributele nonatomice în atribute atomice, proces în care se adaugă mai multe linii pentru fiecare categorie de permis pentru un șofer după cum urmează:

id_angajat#	nume	prenume	..	tip_angajat	id_permis#	categorie	data	ora
20000	Ion	Vasile	..	SOFER	1	A	06-06-2012	12:00
20000	Ion	Vasile	..	SOFER	2	B	06-06-2013	12:00
20000	Ion	Vasile	..	SOFER	3	C	06-06-2017	12:00
20020	Ovidie	Ion	..	SOFER	1	C	06-06-2018	12:00
20100	Constantin	Gheorghe	..	SOFER	1	B	06-06-2010	12:00
20100	Constantin	Gheorghe	..	SOFER	2	C	06-06-2012	12:00

Table 2: Transformarea în FN1

Forma normala 2 - exemplu 1

Pentru ca relația *SOFER_detine_PERMIS* să fie în formă normală 2 (FN2) trebuie să fie mai întâi în FN1, lucru realizat anterior, și în plus trebuie să îndeplinească condiția ca fiecare atribut care nu este cheie (nu participă la cheia primară) este **dependent de întreaga cheie primară**.

Relația de mai sus nu se află în FN2 întrucât se găsesc atributele *nume*, *prenume*, *nr_telefon*, *..*, *tip_angajat* care nu sunt chei și trebuie să depindă direct de întreaga cheie primară **id_angajat#** și **id_permis#**. Atributele

nu depind direct de întreaga cheie primară, deoarece se observă dependența directă dintre $id_angajat\#$ și $nume, prenume, \dots, tip_angajat$, însemnând că $nume, prenume, \dots, tip_angajat$ depind direct doar de o parte a cheii primare (i.e. $id_angajat\#$ nu și de $id_permis\#$).

Aplicăm regula *Casey-Delobel* pentru FN2:

Avem relația $R(K1, K2, X, Y)$, unde $K1=id_angajat\#, K2=id_permis\#$ care definesc cheia primară compusă, iar X și Y sunt mulțimi de attribute, astfel încât $K1 \rightarrow X$, unde $X=\{nume, prenume, nr_telefon, \dots, tip_angajat\}$ și $Y=\{categorie, data, ora\}$.

Din cauza dependenței funcționale $K1 \rightarrow X$ care arată că R nu este în FN2, se înlocuiește R (fără pierdere de informație) prin două proiecții: $(K1, K2, Y)$ și $(K1, X)$.

Astfel avem în final proiecțiile:

- $\{id_angajat\#\} \rightarrow \{nume, prenume, nr_telefon, data_angajare, id_firma, nume_firma, tip_angajat\}$, $id_angajat\#$ determinând funcțional attributele menționate anterior
- $\{id_permis\#, id_angajat\#\} \rightarrow \{categorie, data, ora\}$

$id_angajat\#$	$id_permis\#$	$categorie$	$data$	ora
20000	1	A	06-06-2012	12:00
20000	2	B	06-06-2013	12:00
20000	3	C	06-06-2017	12:00
20020	1	C	06-06-2018	12:00
20100	2	B	06-06-2010	12:00
20100	1	C	06-06-2012	12:00

Table 3: Proiectia $R1(K1, K2, Y)$

$id_angajat\#$	$nume$	$prenume$	\dots	$tip_angajat$
20000	Ion	Vasile	..	SOFER
20020	Ovidie	Ion	..	SOFER
20100	Constantin	Gheorghe	..	SOFER

Table 4: Proiectia $R2(K1, X)$

Forma normala 3 - exemplu 1

Pentru ca relația *SOFER_detine_PERMIS* să fie în formă normală 3 (FN3) trebuie să fie mai întâi în FN2, lucru realizat anterior, și în plus trebuie să îndeplinească condiția ca fiecare atribut care nu este cheie(nu participă la o cheie) să **depindă direct de cheia primară**.

În proiecția $R1(K1, K2, Y)$ toate atributele depind direct de cheia primară. *Categoria* reprezintă tipul de autovehicul pe care permisul îi oferă dreptului unui șofer care l-a obținut să conducă autovehicule de categoria respectivă. Data și ora reprezintă când a fost obținut permisul de o persoană(de acum supranumită șofer). Pentru această relație nu există dependențe tranzitive, de unde rezultă că se află în FN3. Se va denumi această relație *SOFER_detine_PERMIS*.

Se verifică acum relația $R2(K1, X)$.

id_angajat#	nume	prenume	nr_telefon	..	id_firma	nume	tip_firma	tip_angajat
20000	Ion	Vasile	0722123456	..	1100	Road Logistics	TRANSPORT	SOFER
20020	Ovidie	Ion	0727124356	..	1100	Road Logistics	TRANSPORT	SOFER
20100	Constantin	Gheorghe	0735123456	..	1400	Lextom Trans Asd	TRANSPORT	SOFER

Table 5: Relația $R2(K1, X)$

Relația R2 este în FN2 după cum a fost demonstrat anterior, însă nu se află în FN3 întrucât se observă că atributele *nume(firmă)*, *tip_firmă* depind tranzitiv de cheia primară *id_angajat* prin intermediul atributului *id_firma*.

Pentru a reduce relația în FN3 se aplică regula *Casey-Delobel*. Relația se descompune, prin eliminarea dependențelor funcționale tranzitive, în proiecții.

Avem relația $R(K, X_1, X_2, X_3)$, unde $X_2 = \{nume(firma), tip_firma\}$ depinde tranzitiv de $K = id_angajat\#$ (i.e. cheia primară a lui R). Se presupune că $K \rightarrow X_1 \rightarrow X_2$. ($X_1 = id_firma$ și $X_3 = \{nume, prenume, nr_telefon, ..\}$)

Din cauza dependenței funcționale $X_1 \rightarrow X_2$ care arată că R nu este în FN3, se înlocuiește R (**fără pierdere de informație**) prin două proiecții $R1(K, X_1, X_3)$ și $R2(X_1, X_2)$.

- $\{id_angajat\# \} \rightarrow \{nume, prenume, nr_telefon, data_angajare, salariu, id_firma, tip_angajat\}$
- $\{id_firma\# \} \rightarrow \{nume, tip_firma\}$

id_angajat#	nume	prenume	nr_telefon	..	id_firma	tip_angajat
20000	Ion	Vasile	0722123456	..	1100	SOFER
20020	Ovidie	Ion	0727124356	..	1100	SOFER
20100	Constantin	Gheorghe	0735123456	..	1400	SOFER

Table 6: Proiectia $R2-1(K, X_1, X_3)$

id_firma#	nume	tip_firma
1100	Road Logistics	TRANSPORT
1100	Road Logistics	TRANSPORT
1400	Lextom Trans Asd	TRANSPORT

Table 7: Proiectia $R2-2(K_1, K_2)$

Forma normala 1 - exemplu 2

Se dă relația *CAMION_transportă_INVENTAR_MARFĂ_de_la_DEPOZIT_la_DEPOZIT* în care prin intermediul unui camion este transportată marfă, marfă pentru care se face o inventariere a transportului de la un depozit la altul. Se pleacă de la prima variantă a relației:

id_transport#	depozit_plecure#	id_locatie	id_firma	id echipa_paza	depozit_destinatie#	id_locatie	id_firma	id echipa_paza
10000	3000	100	4	1100	500	30	2	100
10010	3000	100	4	1100	500	30	2	100
10020	3500	120	10	1200	1000	50	3	200

id_camion#	id_firma	marca	nr_inmatriculare	id_inventar_transport#	id_lot_marfa#	nume	cantitate	data_plecure	ora_plecure
15	1100	VOLVO	CT12FSD	100 155	100 105	carne miel carne porc	100 300	6-05-2021	08:00
25	1100	VOLVO	CT90MMM	210	110	carne pui	250	8-05-2021	06:00
35	1200	SCANIA	PH13SCN	265	120	rosii	1000	6-05-2021	08:00

Table 8: Relația nu e în FN1

în care se observă că atributelor *id_inventar_transport*, *id_lot_marfa*, *nume(marfa)*, *cantitate* le corespund o listă de valori, i.e. nu le corespund o valoare indivizibilă(atomică), de unde se trage concluzia că relația *CAMION_transportă_INVENTAR_MARFĂ_de_la_DEPOZIT_la_DEPOZIT* nu se află în forma normală 1 (FN1). Se aduce relația în FN1 spărgând attributele nonatomice în attribute atomice, proces în care se adaugă mai multe linii pentru fiecare lot de marfă transportat și detaliile sale după cum urmează:

id_ transport#	depozit_ plecare#	id_ locatie(dp)	id_ firma(dp)	id echipa _paza(dp)	depozit_ destinatia#	id_ locatie(dd)	id_ firma(dd)	id echipa _paza(dd)
10000	3000	100	4	1100	500	30	2	100
10000	3000	100	4	1100	500	30	2	100
10010	3000	100	4	1100	500	30	2	100
10020	3500	120	10	1200	1000	50	3	200

id_ camion#	id_ firma(c)	marca	nr_ inmatriculare	id_inventar _transport#	id_lot _marfa#	nume	cantitate	data_ plecare	ora_ plecare
15	1100	VOLVO	CT12FSD	100	100	carne miel	100	6-05-2021	08:00
15	1100	VOLVO	CT12FSD	155	105	carne porc	300	6-05-2021	08:00
25	1100	VOLVO	CT90MMM	210	110	carne pui	250	8-05-2021	06:00
35	1200	SCANIA	PH13SCN	265	120	rosii	1000	6-05-2021	08:00

Table 9: Transformarea în FN1

Obs: Pentru attributele care se repetă s-a adăugat în paranteză la final un alias, a.î. să fie clar de care cheie depinde.

Forma normala 2 - exemplu 2

Pentru ca relația *CAMION_transportă_INVENTAR_MARFĂ_de_la_DEPOZIT_la_DEPOZIT* să fie în formă normală 2 (FN2) trebuie să fie mai întâi în FN1, lucru realizat anterior, și în plus trebuie să îndeplinească condiția ca fiecare atribut care nu este cheie (nu participă la cheia primară) este **dependent de întreaga cheie primară**.

Relația de mai sus nu se află în FN2 întrucât se găsesc attributele *id_firma(dp)*, *id_firma(dd)*, *id_firma(c)*, *marca*, *nr_inmatriculare* care nu sunt chei și trebuie să depindă direct de întreaga cheie primară **id_transport# + depozit_plecure + depozit_destinatia + id_camion + id_inventar_transport# + id_lot_marfa**. Attributele nu depind direct de întreaga cheie primară, deoarece se observă dependența directă dintre *id_camion* și *id_firma(c)*, *marca*, *nr_inmatriculare* care nu sunt dependente și de *depozit_destinatia* (de exemplu).

Se mai găsesc următoarele dependențe funcționale:

- dependența directă dintre *depozit_plecure* și *id_locatie(dp)*, *id_firma(dp)*, *id echipa_paza(dp)* care nu sunt dependente și de *depozit_destinatia*.
- dependența directă dintre *depozit_destinatia* și *id_locatie(dd)*, *id_firma(dd)*, *id echipa_paza(dd)* care nu sunt dependente și de *depozit_plecure*.

Atributele rămase care nu sunt chei *nume*, *cantitate*, *data_plecare*, *ora_plecare* depind de întreaga cheie primară întrucât se consideră legătura pentru nume de exemplu:

- *id_transport* → ce marfă a fost transportată într-un transport.
- *depozit_plecare* → ce marfă a plecat din depozit.
- *depozit_destinatie* → ce marfă va fi transportată către depozit.
- *id_camion* → ce marfă va conține camionul.
- *id_inventar_transport* → ce marfă face parte din inventarierea transportului.

Analog pentru celelalte atribute.

Aplicăm regula *Casey-Delobel* pentru FN2:

Avem relația $R(K1, K2, K3, K4, K5, K6, X, Y, Z, W)$, unde $K1=id_transport\#$, $K2=depozit_plecare\#$, $K3=depozit_destinatie\#$, $K4=id_camion\#$, $K5=id_inventar_transport\#$, $K6=id_lot_marfa\#$ care definesc cheia primară compusă, iar X, Y, Z și W sunt mulțimi de atribute, astfel încât avem dependențele directe:

- $K2 \rightarrow X$, unde $X = \{id_locatie(dp), id_firma(dp), id_echipa_paza(dp)\}$
- $K3 \rightarrow Y$, unde $Y = \{id_locatie(dd), id_firma(dd), id_echipa_paza(dd)\}$
- $K4 \rightarrow Z$, unde $Z = \{id_firma(c), marca, nr_inmatriculare\}$

În plus avem lista de atribute $W = \{nume, cantitate, data_plecare, ora_plecare\}$.

Din cauza dependențelor funcționale $K2 \rightarrow X$, $K3 \rightarrow Y$ și $K4 \rightarrow Z$ care arată că R nu este în FN2, se înlocuiește R (fără pierdere de informație) prin 4 proiecții: $(K1, K2, K2, K4, K5, K6, Z)$, $(K2, X)$, $(K3, Y)$, $(K4, Z)$.

Astfel avem în final proiecțiile:

- $\{id_transport\#, depozit_plecare\#, depozit_destinatie\#, id_camion\#,$

$\text{id_inventar_transport\#}, \text{id_lot_marfa\#} \rightarrow \{\text{nume}, \text{cantitate}, \text{data_plecare}, \text{ora_plecare}\}$

- $\{\text{depozit_plecare\#}\} \rightarrow \{\text{id_locatie}, \text{id_firma}, \text{id_echipa_paza}\}$
- $\{\text{depozit_destinatie\#}\} \rightarrow \{\text{id_locatie}, \text{id_firma}, \text{id_echipa_paza}\}$
- $\{\text{id_camion\#}\} \rightarrow \{\text{id_firma}, \text{marca}, \text{nr_inmatriculare}\}$

id_ transport#	depozit_ plecare#	depozit_ destinatie#	id_ camion#	id_inventar_ transport#	id_lot_ marfa#	nume	cantitate	data_ plecare	ora_ plecare
10000	3000	500	15	100	100	carne miel	100	6-05-2021	08:00
10000	3000	500	15	155	105	carne porc	300	6-05-2021	08:00
10010	3000	500	25	210	110	carne pui	250	8-05-2021	06:00
10020	3500	1000	35	265	115	rosii	1000	6-05-2021	08:00

Table 10: Proiectia $R1(K_1, K_2, K_3, K_4, K_5, K_6, W)$

depozit_plecare#	id_locatie	id_firma	id_echipa_paza
3000	100	4	1100
3500	120	10	1200

Table 11: Proiectia $R2(K_2, X)$

depozit_destinatie#	id_locatie	id_firma	id_echipa_paza
500	30	2	100
1000	50	3	200

Table 12: Proiectia $R3(K_3, Y)$

id_camion	id_firma	marca	nr_inmatriculare
15	1100	VOLVO	CT12FSD
25	1100	VOLVO	CT90MMM
35	1200	SCANIA	PH13SCN

Table 13: Proiectia $R4(K_4, Z)$

Pentru relația *transport* se observă că a ajuns la 6 chei primare, lucru ineficient în practică, fiind o decizie teribilă, motiv pentru care se înlocuiește cheia primară compusă cu o cheie primară simplă formată doar din *id_transport*.

Forma normală 3 - exemplu 2

Relația anterioară se află în forma normală 3 (FN3) dacă se află întâi în FN2, lucru realizat anterior, și în plus trebuie să îndeplinească condiția ca fiecare

atribut care nu este cheie(nu participă la o cheie) **să depindă direct de cheia primară.**

Relațiile 2-4 prezentate anterior prin proiecții se află deja în FN3. Se verifică acum relația 1: aceasta este în FN2, transformare realizată anterior, dar nu este în FN3, deoarece se observă că attributele *nume* și *cantitate* depind tranzitiv de cheia primară *id_transport* prin intermediul atributului *id_lot_marfa*, atribut care depinde și el tranzitiv prin intermediul atributului *id_inventar_transport*.

Pentru a reduce relația în FN3 se aplică regula *Casey-Delobel*. Relația se descompune, prin eliminarea dependențelor funcționale tranzitive, în proiecții.

Avem relația $R(K, X_1, X_2, X_3, X_4)$, unde K este cheia primară *id_transport*, $X_1 = id_inventar_transport$, $X_2 = id_lot_marfa$, $X_3 = \{nume, cantitate\}$ și $X_4 = \{depozit_plecare, depozit_destinatie, data_plecare, ora_plecare\}$. Sunt evidente dependențele tranzitive $K \rightarrow X_1 \rightarrow X_2$ și $X_1 \rightarrow X_2 \rightarrow X_3$.

Din cauza acestor dependențe funcționale relația nu se află în FN3, motiv pentru care se înlocuiește R (**fără pierdere de informație**) prin 3 proiecții $R1(K, X_4)$, $R2(X_1, X_2, K)$ și $R3(X_2, X_3)$. În crearea proiecțiilor s-a ținut cont și de cardinalitățile dintre entități.

- $\{id_transport\# \} \rightarrow \{depozit_plecare, depozit_destinatie, id_camion, data_plecare, ora_plecare\}$
- $\{id_inventar_transport\#, id_lot_marfa\# \} \rightarrow \{id_transport\}$
- $\{id_lot_marfa\# \} \rightarrow \{nume, cantitate\}$

id_transport#	depozit_plecure#	depozit_destinatie#	id_camion#	id_inventar_transport	id_lot_marfa#	data_plecure	ora_plecure
10000	3000	500	15	100	100	6-05-2021	08:00
10000	3000	500	15	155	105	6-05-2021	08:00
10010	3000	500	25	210	110	8-05-2021	06:00
10020	3500	1000	35	265	115	6-05-2021	08:00

Table 14: Proiectia $R1-1(K, X_4)$

id_inventar_transport#	id_lot_marfa	id_transport
100	100	10000
155	105	10000
210	110	10010
265	115	10020

Table 15: Proiectia $R1-2(X_1, X_2, K)$

id_lot_marfa#	nume	cantitate
100	carne miel	100
105	carne porc	300
110	carne pui	250
115	rosii	1000

Table 16: Proiectia $R1-3(X_2, X_3)$

Obs: Datele din următoarele tabele/exemple sunt fictive, nu sunt prezente în baza de date creată. S-a ales această metodă pentru o exemplificare mai bună per exemplu. De asemenea vom avea convenția de a adăuga înainte de fiecare cheie primară un string care să semnifice pentru ce entitate este desemnată cheia primară.

Forma normala Boyce-Codd

Se consideră că o relație R se află în forma normală **Boyce-Codd** dacă și numai dacă fiecare determinant este o cheie candidat. Un determinant reprezintă un atribut sau o mulțime de attribute neredundante, care constituie un identificator unic pentru un alt atribut sau o altă mulțime de date.

Se dă relația *SOFER_CONDUCE_CAMION* în care un șofer poate conduce mai multe camioane care o preluăm din forma normală 1.

id_historic_camioane_conduse#	id_angajat#	id_camion#	id_firma	marca	nr_inmatriculare	data_inceput	data_sfarsit
ICC1	A1	C1	F1	DAF	PX13ASD	06-06-2019	07-07-2019
ICC2	A2	C2	F1	DAF	PX21QWE	12-04-2019	11-11-2019
ICC3	A3	C3	F2	MAN	PX14HGA	06-05-2019	01-12-2019
ICC3	A4	C4	F3	DAF	PX13ICC	06-01-2019	07-02-2019

Table 17: Relatia *SOFER_CONDUCE_CAMION*

În relația anterioară se găsește determinantul $X=nr_inmatriculare$ care îl determină funcțional pe $K3=id_camion\#$ și $Y=\{id_firma, marca\}$, X fiind

cheie candidat. Astfel aplicăm regula Casey Delobel pentru $R(K1\#, K2\#, K3\#, X, Y, Z)$, unde $K1=id_istoric_camioane_conduse\#$, $K2=id_angajat\#$, $Z=\{data_inceput, data_sfarsit\}$. Astfel se ajunge la proiecțiile: $R1(K1\#, K2\#, X, Z)$ și $R2(X\#, K3)$ care au adus relația în *BCNF*.

id_istoric camioane_conduse#	id_ angajat#	nr_ inmatriculare	data_ inceput	data_ sfarsit
ICC1	A1	PX13ASD	06-06-2019	07-07-2019
ICC2	A2	PX21QWE	12-04-2019	11-11-2019
ICC3	A3	PX14HGA	06-05-2019	01-12-2019
ICC3	A4	PX13ICC	06-01-2019	07-02-2019

Table 18: $R1(K1\#, K2\#, X, Z)$

nr_ inmatriculare#	id_ camion	marca	id_ firma
PX13ASD	C1	DAF	F1
PX21QWE	C2	DAF	F1
PX14HGA	C3	MAN	F2
PX13ICC	C4	DAF	F3

Table 19: $R2(X\#, K3)$

Analog se procedează și pentru relația *transport*. (În exemplu se vor trece doar atributele de folos pentru exemplul dat)

id_depozit _plecare#	id_depozit _destinatie#	id_lot _marfa#	id_ camion#	nr_ inmatriculare
D1	D5	LM1	C1	PX13ASD
D1	D6	LM2	C1	PX13ASD
D2	D5	LM1	C1	PX13ASD
D2	D6	LM2	C1	PX13ASD
D1	D3	LM5	C3	PX14HGA
D1	D3	LM4	C4	PX13ICC

Table 20: Relația *transport*

Se găsește analog determinantul $X=nr_inmatriculare$ care determină funcțional pe $K4=id_camion\#$. Se mai fac notațiile $K1=id_depozit_plecare\#$, $K2=id_depozit_destinatie\#$, $K3=id_lot_marfa\#$. Astfel se ajunge la proiecțiile $R1(K1\#, K2\#, K3\#, X)$ și $R2(X\#, K4)$.

id_depozit _plecare#	id_depozit _destinatie#	id_lot _marfa#	nr_ inmatriculare
D1	D5	LM1	PX13ASD
D1	D6	LM2	PX13ASD
D2	D5	LM1	PX13ASD
D2	D6	LM2	PX13ASD
D1	D3	LM5	PX14HGA
D1	D3	LM4	PX13ICC

Table 21: Relația $R1(K1\#, K2\#, K3\#, X)$

nr_ inmatriculare#	id_ camion
PX13ASD	C1
PX13ASD	C1
PX13ASD	C1
PX13ASD	C1
PX14HGA	C3
PX13ICC	C4

Table 22: Relația $R2(X\#, K4)$

Forma normală 4

O relație se află în formă normală 4 (FN4) dacă și numai dacă relația este în BCNF și nu conține relații m:n independente.

Se dă relația anterior transformată(*transport*) în BCNF $R(id_depozit_destinatie\#, id_depozit_plecare\#, id_lot_marfa\#, nr_inmatriculare\#)$

id_depozit _plecare#	id_depozit _destinatie#	id_lot _marfa#	nr_ inmatriculare#
D1	D5	LM1	PX13ASD
D1	D6	LM2	PX13ASD
D2	D5	LM1	PX13ASD
D2	D6	LM2	PX13ASD

Table 23: Relația *transport*

în care se găsesc următoarele dependențe multiple:

- $nr_inmatriculare \twoheadrightarrow id_depozit_plecare$

- $nr_inmatriculare \rightarrow \rightarrow id_depozit_destinatie$
- $nr_inmatriculare \rightarrow \rightarrow id_lot_marfa$

și se obțin următoarele rezultate după aplicarea lui FN4:

- $transport1 = R1(nr_inmatriculare\#, id_depozit_plecare)$
- $transport2 = R2(nr_inmatriculare\#, id_depozit_destinatie)$
- $transport3 = R3(nr_inmatriculare\#, id_lot_marfa)$

cu următoarele:

nr_ inmatriculare#	id.depozit _plecare#
PX13ASD	D1
PX13ASD	D2

Table 24: Relația *transport1*

nr_ inmatriculare#	id.depozit _destinatie#
PX13ASD	D5
PX13ASD	D6

Table 25: Relația *transport2*

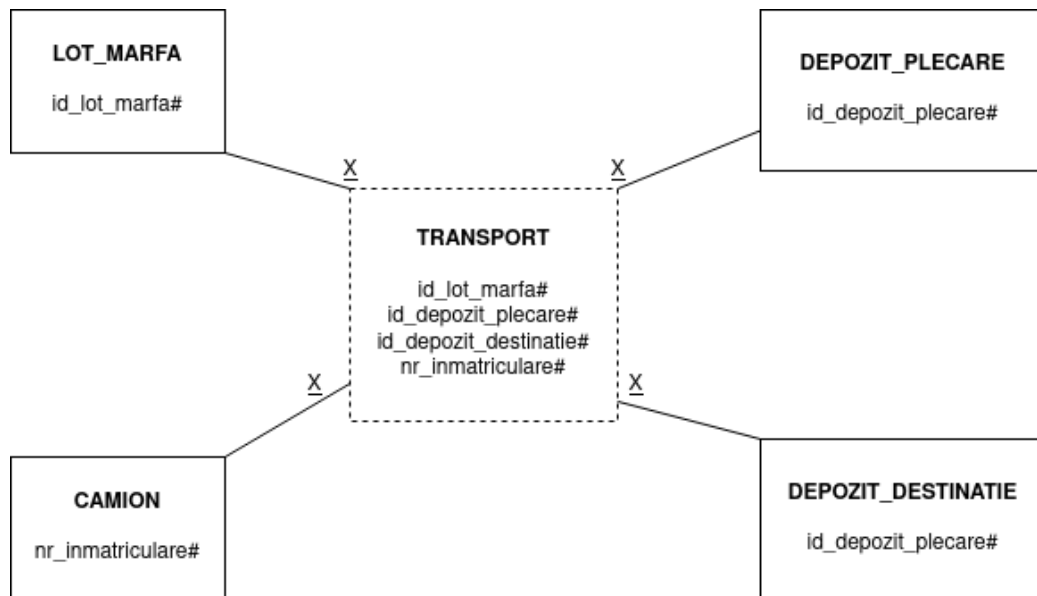
nr_ inmatriculare#	id.lot_ marfa#
PX13ASD	LM1
PX13ASD	LM2

Table 26: Relația *transport3*

Forma normală 5

O relație se află în formă normală 5 (FN5) dacă și numai dacă se află în FN4 și nu conține dependențe ciclice.

Se dă exemplul următor: un lot de marfă poate fi transportat de la mai multe depozite la mai multe depozite și poate să fie transportat de mai multe camioane. Aceasta relație este una de tip 3 și are reprezentarea:



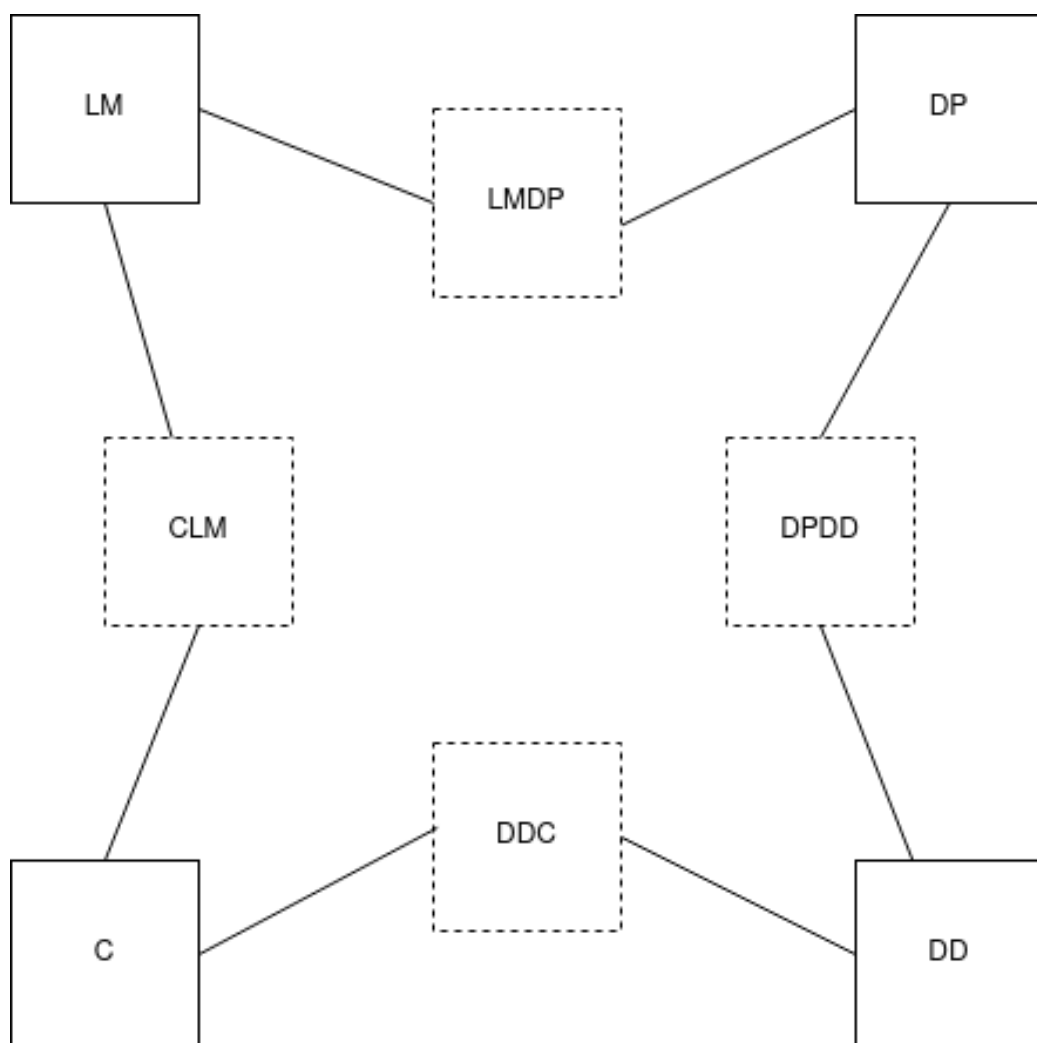
Relație tip 3

cu posibilele înregistrări:

id_depozit_plecara#	id_depozit_destinatie#	id_lot_marfa#	nr_inmatriculare#
D1	D5	LM1	PX13ASD
D1	D6	LM2	PX13ASD
D2	D5	LM1	PX13ASD
D2	D6	LM2	PX13ASD

Table 27: Relația *transport*

Această relație de tip 3 poate fi echivalentă cu 4 relații de tip 2 doar dacă aceste relații de tip 2 sunt ciclice.



Relatie tip 2

Relația fiind ciclică, atunci când se va face o interogare cu toate join-urile se va obține un rezultat echivalent cu cel obținut din interogarea relației de tip 3. Anterior au fost menționate condițiile ca o relație să fie în FN5, printre care și condiția de a nu conține dependențe ciclice, motiv pentru care cele 3 relații de tip 2 compun o diagramă care conține dependențe ciclice, deci relația nu se află în FN5. În schimb relația de tip 3 se află în FN5.

10 Denormalizare

Denormalizarea presupune mărirea redundanței pentru a reduce numărul de join-uri care trebuie efectuate pentru rezolvarea unei interogări, astfel se micșorează timpul de execuție.

→Exemplu când denormalizarea nu este utilă:

În tabelul *LOT_MARFA* în care sunt stocate loturile de marfă din transporturile efectuate. Loturile de marfă au atributele **nume** și **cantitate** care pot conține valori repetitive, aceeași cantitate putând aparținând mai multor loturi de marfă, analog pentru nume. Dacă în baza de date există un tabel separat pentru cantitate și unul pentru nume, fiecare împreună cu id-ul lotului de marfă căruia îi corespunde, este necesar să se aplice procesul de denormalizare în urma căruia atributele nume și cantitate se vor plasa în tabelul *LOT_MARFA* întrucât ar fi inefficient ca în interogări aceste atribute să se afle în tabele separate datorită unui dublu join care ar mări timpul de execuție în cazul în care se dorește aflarea tuturor informațiilor despre un anumit lot de marfă.

→Exemplu când denormalizarea este utilă: În tabelul *ANGAJAT* există atributul **id_firma** care reprezintă o cheie externă pentru o cheie primară din tabelul *FIRMA*, acest atribut desemnând la ce firmă lucrează un angajat. Cu toate acestea se poate afla la ce firmă lucrează un angajat dacă se realizează un join între *ANGAJAT*, *ISTORIC_CAMIOANE_CONDUSE*, *CAMION* și *FIRMA* pentru un angajat de tip șofer și un join între *ANGAJAT*, *ECHIPA_PAZA* și *FIRMA*. Este deja evident că pentru aflarea acestei informații (la ce firmă lucrează un angajat/ce angajați lucrează la o firmă) o interogare ar avea un număr de join-uri prea mare pentru ce informație se dorește. Astfel se realizează denormalizarea acestui tabel prin adăugarea atributului **id_firma** pentru a reduce nr. de join-uri din posibilele interogări dorite, lucru care rezultă într-un timp de execuție mai mic. Astfel această denormalizare se dovedește utilă.

11 Optimizarea unei cereri - ex17

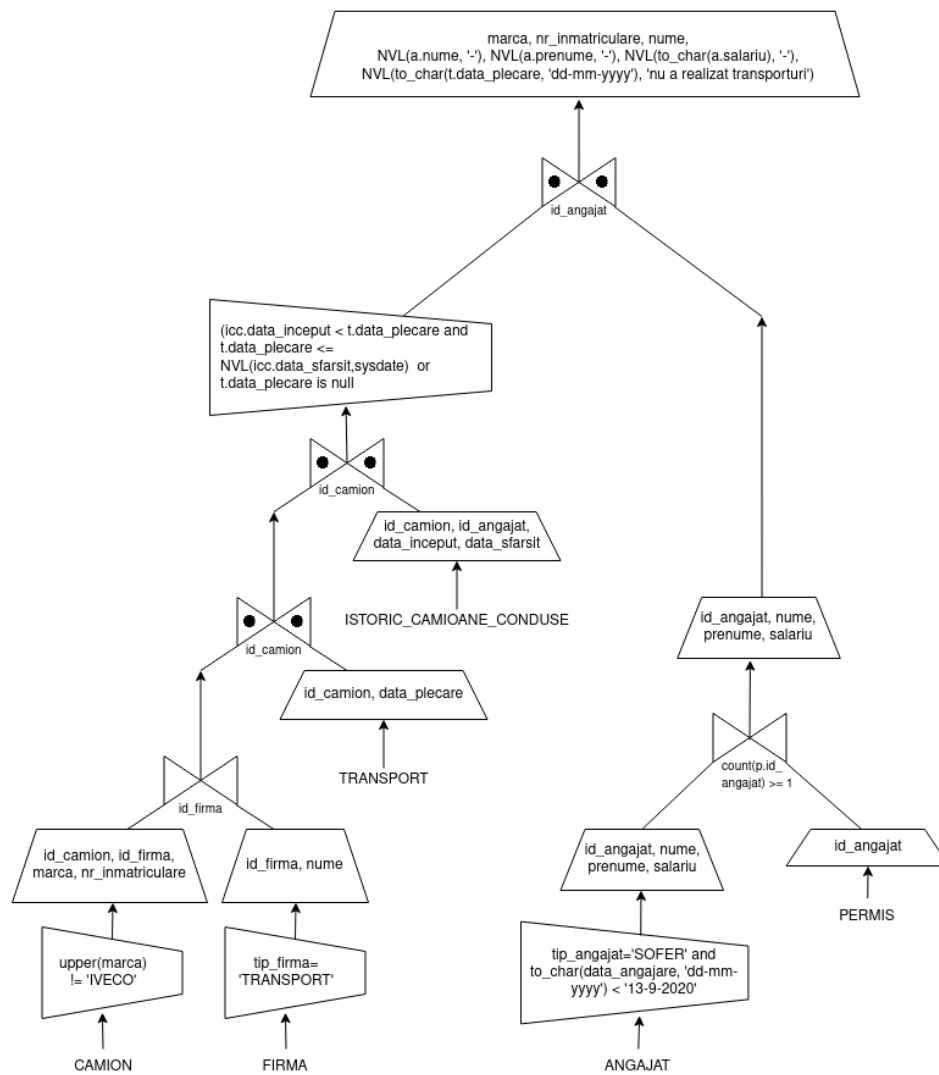
Se cer următoarele informații:

Pentru fiecare camion să se afișeze marca, nr. de înmatriculare, numele firmei, de ce șoferi a fost condus (pentru fiecare șofer să se precizeze numele, prenumele și salariul, în cazul în care un camion nu a fost condus de un șofer să se înlocuiască aceste coloane cu '-'), dar și data de plecare pentru fiecare transport realizat de șofer cu acest camion. Dacă unui angajat i-a fost alocat un camion dar nu a realizat niciun transport în loc de dată să fie afișat mesajul 'nu a realizat transporturi'. Să se afișeze informațiile despre angajați doar dacă sunt șoferi, au fost angajați înainte de data de 13 sept 2020 și dețin cel puțin un permis auto, în cazul în care un camion a fost condus de un șofer care nu îndeplinește aceste cerințe se vor înlocui coloanele cu informații despre angajat cu '-'. Să se excludă camioanele de marcă IVECO.

Prima cerere *SQL* implementată:

```
01 | WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin cel mult
    | ↪ 2 permise auto
02 | (SELECT p.id_angajat, a.num, a.prenume, a.salariu
03 | FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
04 | WHERE a.tip_angajat = 'SOFER'
05 | and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
06 | GROUP BY p.id_angajat, a.num, a.prenume, a.salariu
07 | HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1 permis
08 | )
09 |
10 | SELECT c.marca, c.nr_inmatriculare, f.num,
11 | NVL(a.num, '-') "Nume sofer", NVL(a.prenume, '-') "Prenume sofer",
12 | NVL(to_char(a.salariu), '-') "Salariu sofer",
13 | NVL(to_char(t.data_plecure, 'dd-mm-yyyy'), 'nu a realizat
    | ↪ transporturi') "Data transport"
14 | FROM camion c JOIN firma f ON (c.id_firma = f.id_firma)
15 | FULL OUTER JOIN istoric_camioane_conduse icc ON (c.id_camion =
    | ↪ icc.id_camion)
16 | FULL OUTER JOIN ang a ON (icc.id_angajat = a.id_angajat)
17 | FULL OUTER JOIN transport t ON (c.id_camion = t.id_camion)
18 | WHERE
19 | upper(c.marca) != 'IVECO' and
20 | f.tip_firma = 'TRANSPORT' and
21 | ((icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(icc.
    | ↪ data_sfarsit, sysdate)
22 | ) or t.data_plecure is null
23 | );
```

S-a realizat următorul arbore algebric asociat cererii:



Arbore v1

și expresia algebrică asociată:

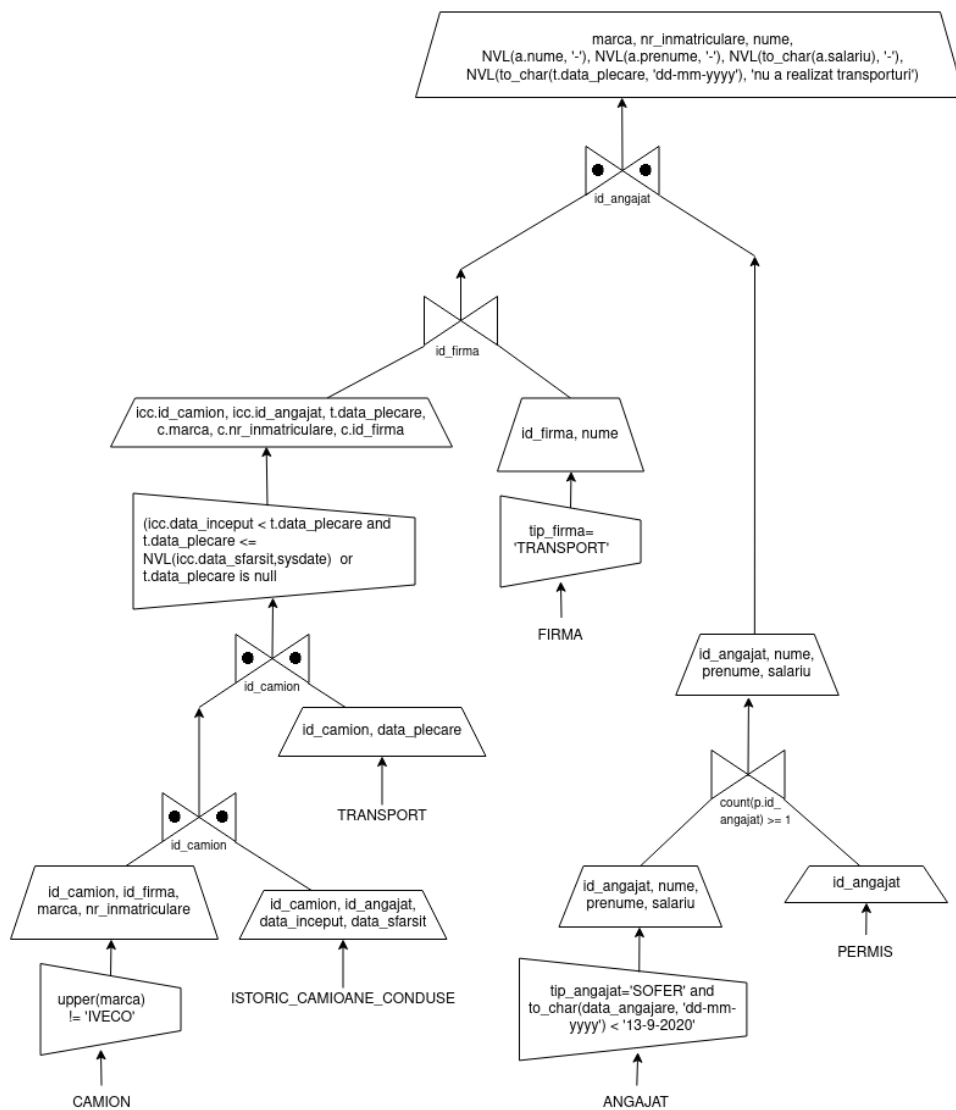
```
01 | R1=SELECT(CAMION, upper(marca)!='IVECO')
02 | R2=PROJECT(R1, id_camion, id_firma, marca, nr_inmatriculare}
03 | R3=SELECT(FIRMA, tip_firma='TRANSPORT')
04 | R4=PROJECT(R3, id_firma, nume)
05 | R5=JOIN(R2, R4, id_firma)
06 | R6=PROJECT(TRANSPORT, id_camion, data_plecure)
07 | R7=FULL OUTER JOIN(R5, R6, id_camion)
08 | R8=PROJECT(ISTORIC_CAMIOANE_CONDUSE, id_camion, id_angajat, data_inceput,
    ↪ data_sfarsit)
09 | R9=FULL OUTER JOIN(R7, R8, id_camion)
10 | R10=SELECT(R9, (icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(
    ↪ icc.data_sfarsit, sysdate)) or t.data_plecure is null)
11 | R11=SELECT(ANGAJAT, tip_angajat = 'SOFER' and to_char(data_angajare, 'dd-mm-
    ↪ yyyy') < '13-9-2020')
12 | R12=PROJECT(R11, id_angajat, nume, prenume, salariu)
13 | R13=PROJECT(PERMIS, id_angajat)
14 | R14=JOIN(R12, R13, count(p.id_angajat) >= 1)
15 | R15=PROJECT(R14, id_angajat, nume, prenume, salariu)
16 | R16=FULL OUTER JOIN(R10, R15, id_angajat)
17 | Rez=R17=PROJECT(R16, marca, nr_inmatriculare, nume, NVL(nume, '-'), NVL(
    ↪ prenume, '-'), NVL(to_char(salariu), '-'), NVL(to_char(t.data_plecure
    ↪ , 'dd-mm-yyyy'), 'nu a realizat transporturi'))
```

Se observă faptul că deja o parte a arborelui este optimizată folosind proprietatea de compunere a proiecțiilor și de compunere a selecțiilor.

Se fac următoarele optimizări:

- se folosește regula de optimizare 1 care spune că selecțiile se execută cât mai devreme posibil pentru R10, selecție care ar putea fi realizată mai devreme, totodată se va folosi și proprietatea 2 care spune că operațiile de join și produs cartezian sunt asociative. În alegerea primelor join-uri vom folosi și regula de optimizare 3 care spune că dacă există mai multe join-uri atunci cel care se execută primul este cel mai restrictiv. Astfel se va face un *FULL OUTER JOIN* între *CAMION* și *ISTORIC_CAMIOANE_CONDUSE* după un alt *FULL OUTER JOIN* între rezultatul anterior și *TRANSPORT* după care vom face selecția care anterior era R10 și o proiecție pentru a elimina attributele nefolositoare.

Și se ajunge la următorul arborele algebric:



Arbore v2 optimizat

și expresia algebrică asociată:

```

01 | R1=SELECT(CAMION, upper(marca)!='IVECO')
02 | R2=PROJECT(R1, id_camion, id_firma, marca, nr_inmatriculare}
03 | R3=PROJECT(ISTORIC_CAMIOANE_CONDUSE, id_camion, id_angajat, data_inceput,
    ↳ data_sfarsit)
04 | R4=FULL OUTER JOIN(R2, R3, id_camion)
05 | R5=PROJECT(TRANSPORT, id_camion, data_plecure)
06 | R6=FULL OUTER JOIN(R4, R5, id_camion)
07 | R7=SELECT(R6, (icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(
    ↳ icc.data_sfarsit, sysdate)) or t.data_plecure is null)
08 | R8=PROJECT(R7, icc.id_camion, icc.id_angajat, t.data_plecure, c.marca, c.
    ↳ nr_inmatriculare, c.id_firma)
09 | R9=SELECT(FIRMA, tip_firma='TRANSPORT')
10 | R10=PROJECT(R9, id_firma, nume)
11 | R11=JOIN(R8, R10, id_firma)
12 | R12=SELECT(ANGAJAT, tip_angajat = 'SOFER' and to_char(data_angajare, 'dd-mm-
    ↳ yyyy') < '13-9-2020')
13 | R13=PROJECT(R12, id_angajat, nume, prenume, salariu)
14 | R14=PROJECT(PERMIS, id_angajat)
15 | R15=JOIN(R13, R14, count(p.id_angajat) >= 1)
16 | R16=PROJECT(R15, id_angajat, nume, prenume, salariu)
17 | R17=FULL OUTER JOIN(R11, R16, id_angajat)
18 | Rez=R18=PROJECT(R17, marca, nr_inmatriculare, nume, NVL(numa, '-'), NVL(
    ↳ prenume, '-'), NVL(to_char(salariu), '-'), NVL(to_char(t.data_plecure
    ↳ , 'dd-mm-yyyy'), 'nu a realizat transporturi'))

```

Cererea optimizată finală este:

```

01 | WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin cel mult
    ↳ 2 permise auto
02 | (SELECT p.id_angajat, a.nume, a.prenume, a.salariu
03 | FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
04 | WHERE a.tip_angajat = 'SOFER'
05 | and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
06 | GROUP BY p.id_angajat, a.nume, a.prenume, a.salariu
07 | HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1 permis
08 | ),
09 | tc AS
10 | (SELECT icc.id_camion, icc.id_angajat, t.data_plecure, c.marca, c.
    ↳ nr_inmatriculare, c.id_firma
11 | FROM camion c FULL OUTER JOIN istoric_camioane_conduse icc ON (c.
    ↳ id_camion = icc.id_camion)
12 | FULL OUTER JOIN transport t ON (c.id_camion = t.
    ↳ id_camion)
13 | WHERE c.marca != 'IVECO' and
14 | (t.data_plecure is null or icc.data_inceput is null or
15 | (icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(
    ↳ icc.data_sfarsit, sysdate))
16 | )
17 | )
18 |
19 | SELECT tc.marca, tc.nr_inmatriculare, f.nume,
20 | NVL(a.nume, '-') "Nume sofer", NVL(a.prenume, '-') "Prenume sofer",
21 | NVL(to_char(a.salariu), '-') "Salariu sofer",
22 | NVL(to_char(tc.data_plecure, 'dd-mm-yyyy'), 'nu a realizat
    ↳ transporturi') "Data transport"
23 | FROM tc JOIN firma f ON (tc.id_firma = f.id_firma)
24 | FULL OUTER JOIN ang a ON (tc.id_angajat = a.id_angajat)
25 | WHERE f.tip_firma = 'TRANSPORT';

```

12 Cod SQL pentru crearea tabelelor - ex10

```
01 | -- crearea tabelelor
02 |
03 | CREATE TABLE LOCATIE
04 |     (id_locatie number(10) constraint pk_loc primary key,
05 |      judet varchar2(25),
06 |      localitate varchar2(25),
07 |      strada varchar2(25),
08 |      nr number(5)
09 |     );
10 |
11 | CREATE TABLE POPAS
12 |     (id_popas number(10) constraint pk_pop primary key,
13 |      id_locatie number(10) constraint fk_pop_loc references
14 |          LOCATIE(id_locatie) on delete set null,
15 |      nume varchar2(25),
16 |      tip_popas varchar2(10) constraint tip_pop not null,
17 |      constraint verific_tip_pop check(tip_popas = 'BENZINARIE'
18 |          or tip_popas = 'MOTEL' or tip_popas = '
19 |      → RESTAURANT')
20 |     );
21 |
22 | CREATE TABLE FIRMA
23 |     (id_firma number(12) constraint pk_firma primary key,
24 |      nume varchar2(25) constraint uni_nume_firma unique,
25 |      tip_firma varchar2(10) constraint tip_firma not null,
26 |      constraint verific_tip_firma check(tip_firma = 'CLIENT' or tip_firma = '
27 |      → PAZA'
28 |          or tip_firma = 'TRANSPORT')
29 |     );
30 |
31 | CREATE TABLE ECHIPA_PAZA
32 |     (id_echipa_paza number(5) constraint pk_epaza primary key,
33 |      id_firma number(12) constraint fk_epaza_firma references
34 |          FIRMA(id_firma) on delete cascade
35 |     );
36 |
37 | CREATE TABLE DEPOZIT
38 |     (id_depozit number(5) constraint pk_dep primary key,
39 |      id_locatie number(5) constraint fk_dep_loc references
40 |          LOCATIE(id_locatie) on delete set null,
41 |      id_echipa_paza number(5) constraint fk_dep_epaza
42 |          references ECHIPA_PAZA(id_echipa_paza) on delete set null,
43 |      id_firma number(12) constraint fk_dep_firma references FIRMA(id_firma)
44 |          on delete set null
45 |     );
46 |
47 | CREATE TABLE CAMION
48 |     (id_camion number(5) constraint pk_cam primary key,
49 |      marca varchar2(10),
50 |      nr_inmatriculare varchar2(10) constraint uni_nr_ing_cam unique,
51 |      id_firma number(12) constraint fk_cam_firma references
52 |          FIRMA(id_firma) on delete set null
53 |     );
54 |
55 | CREATE TABLE TRANSPORT
56 |     (id_transport number(10) constraint pk_trans primary key,
57 |      depozit_plecure number(5) constraint fk_trans_dep_p references
58 |          DEPOZIT(id_depozit) on delete cascade,
59 |      depozit_destinatie number(5) constraint fk_trans_dep_d references
60 |          DEPOZIT(id_depozit) on delete cascade,
```

```

59 |         id_camion number(5) constraint fk_trans_cam references
60 |             CAMION(id_camion) on delete cascade,
61 |         data_plecare date default sysdate
62 |     );
63 |
64 | CREATE TABLE LOT_MARFA
65 |     (id_lot_marfa number(10) constraint pk_lotm primary key,
66 |     nume varchar2(25) constraint nume_lotm not null,
67 |     cantitate number(8, 2) constraint cant_lotm not null
68 |     );
69 |
70 | CREATE TABLE INVENTAR_DEPOZIT
71 |     (id_inventar_depozit number(10) constraint pk_inv_dep primary key,
72 |     id_depozit number(5) constraint fk_invdep_dep references
73 |         DEPOZIT(id_depozit) on delete cascade,
74 |     id_lot_marfa number(10) constraint fk_invdep_lotm references
75 |         LOT_MARFA(id_lot_marfa) on delete cascade,
76 |     data_sosire date default sysdate,
77 |     data_plecare date
78 |     );
79 |
80 | CREATE TABLE INVENTAR_TRANSPORT
81 |     (id_inventar_transport number(10),
82 |     id_lot_marfa number(10) constraint fk_invtran_lotm references
83 |         LOT_MARFA(id_lot_marfa) on delete cascade,
84 |     id_transport number(10) constraint fk_invtran_tran references
85 |         TRANSPORT(id_transport) on delete cascade,
86 |     constraint pk_inv_tran primary key(id_inventar_transport, id_lot_marfa)
87 |     );
88 |
89 | CREATE TABLE ANGAJAT
90 |     (id_angajat number(10) constraint pk_ang primary key,
91 |     nume varchar(25),
92 |     prenume varchar(25),
93 |     nr_telefon varchar(14),
94 |     data_angajare date default sysdate,
95 |     salariu number(10),
96 |     id_firma number(12),
97 |     tip_angajat char(6) constraint tip_ang_nn not null,
98 |     id echipa_paza number(5),
99 |     constraint verf_tip check(tip_angajat = 'PAZNIC' or tip_angajat = '
    ↳ SOFER'),
100 |     constraint fk_ang_epaza foreign key(id echipa_paza)
101 |         references ECHIPA_PAZA(id echipa_paza) on delete set null,
102 |     constraint fk_ang_firma foreign key(id_firma)
103 |         references FIRMA(id_firma) on delete set null
104 |     );
105 |
106 | CREATE TABLE ISTORIC_CAMIOANE_CONDUSE
107 |     (id_istoric_camioane_conduce number(10) constraint pk_ist_cam primary
    ↳ key,
108 |     id_angajat number(10) constraint fk_icam_ang references
109 |         ANGAJAT(id_angajat) on delete cascade,
110 |     id_camion number(5) constraint fk_icam_cam references
111 |         CAMION(id_camion) on delete cascade,
112 |     data_inceput date default sysdate,
113 |     data_sfarsit date
114 |     );
115 |
116 | CREATE TABLE PERMIS
117 |     (id_permis number(10),
118 |     id_angajat number(10) constraint fk_perm_ang references

```

```

119 |          ANGAJAT(id_angajat) on delete cascade,
120 |      categorie char(3),
121 |      data date default sysdate,
122 |      constraint pk_perm primary key(id_permis, id_angajat),
123 |      constraint verf_catg check(categorie in ('AM', 'A1', 'A2', 'A', 'B1', '
↪ B',
124 |      'BE', 'C1', 'C1E', 'C', 'CE', 'D1', 'D1E', 'D', 'DE', 'Tr', 'Tb', '
↪ Tv'))
125 |      );
126 |
127 | CREATE TABLE ISTORIC_POPASURI
128 |     (id_istoric_popas number(20),
129 |      id_popas number(10) constraint fk_istpop_pop references POPAS(id_popas)
↪ ,
130 |      id_angajat number(10) constraint fk_istpop_ang references ANGAJAT(
↪ id_angajat),
131 |      data_sosire date default sysdate,
132 |      data_plecare date default sysdate,
133 |      constraint pk_ist_pop primary key(id_istoric_popas, id_popas,
↪ id_angajat, data_sosire)
134 |      );

```


13 Inserarea de date in tabel + utilizarea secventelor - ex 10 + 13

```
01 | /* pentru tabelul Locatie se creeaza urmatoarea secventa SEQ_LOC pentru
    | ↳ inserarea
02 | * inregistrarilor in tabelul Locatie, se va proceda analog pentru celelalte
03 | * exemple fara a se mai specifica intr-un comentariu */
04 | CREATE SEQUENCE SEQ_LOC
05 | INCREMENT BY 10
06 | START WITH 10
07 | MAXVALUE 1000000000
08 | NOCYCLE;
09 |
10 | /* se insereaza date in tabelul Locatie folosind secventa anterior creata
11 | * utilizandu-se metoda implicita | se va proceda analog pentru inserarea de
12 | * date in celelalte tabele fara a se mai specifica intr-un comentariu */
13 | INSERT INTO LOCATIE
14 | VALUES (SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Libertatii', 13);
15 |
16 | INSERT INTO LOCATIE
17 | VALUES (SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Lalelelor', 142);
18 |
19 | INSERT INTO LOCATIE
20 | VALUES (SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Primaverii', 412);
21 |
22 | INSERT INTO LOCATIE
23 | VALUES (SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Unirii', 9991);
24 |
25 | INSERT INTO LOCATIE
26 | VALUES (SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Mihai Eminescu', 991);
27 |
28 | INSERT INTO LOCATIE
29 | VALUES (SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Teilor', 131);
30 |
31 | INSERT INTO LOCATIE
32 | VALUES (SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Teilor', 21);
33 |
34 | INSERT INTO LOCATIE
35 | VALUES (SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Victoriei', 123);
36 |
37 | INSERT INTO LOCATIE
38 | VALUES (SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Zorilor', 13);
39 |
40 | INSERT INTO LOCATIE
41 | VALUES (SEQ_LOC.NEXTVAL, 'Tulcea', 'Tulcea', 'Unirii', 213);
42 |
43 | INSERT INTO LOCATIE
44 | VALUES (SEQ_LOC.NEXTVAL, 'Tulcea', 'Tulcea', 'Victoriei', 113);
45 |
46 | INSERT INTO LOCATIE
47 | VALUES (SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Libertatii', 112);
48 |
49 | INSERT INTO LOCATIE
50 | VALUES (SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', '1 mai', 1213);
51 |
52 | INSERT INTO LOCATIE
53 | VALUES (SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Muncii', 1412);
54 |
55 | INSERT INTO LOCATIE
56 | VALUES (SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Unirii', 212);
57 |
```

```

58 | CREATE SEQUENCE SEQ_POP
59 | INCREMENT BY 20
60 | START WITH 100
61 | MAXVALUE 1000000000
62 | NOCYCLE;
63 |
64 | INSERT INTO POPAS
65 | VALUES (SEQ_POP.NEXTVAL, 10, 'Motel Sunday', 'MOTEL');
66 |
67 | INSERT INTO POPAS
68 | VALUES (SEQ_POP.NEXTVAL, 20, 'Restaurant Ceptura', 'RESTAURANT');
69 |
70 | INSERT INTO POPAS
71 | VALUES (SEQ_POP.NEXTVAL, 40, 'Restaurant Grand', 'RESTAURANT');
72 |
73 | INSERT INTO POPAS
74 | VALUES (SEQ_POP.NEXTVAL, 60, 'Restaurant Unirea', 'RESTAURANT');
75 |
76 | INSERT INTO POPAS
77 | VALUES (SEQ_POP.NEXTVAL, 70, 'Conacul dintre vii', 'MOTEL');
78 |
79 | CREATE SEQUENCE SEQ_FIRMA
80 | INCREMENT BY 100
81 | START WITH 100
82 | MAXVALUE 1000000
83 | NOCYCLE;
84 |
85 | INSERT INTO FIRMA
86 | VALUES (SEQ_FIRMA.NEXTVAL, 'Lidl', 'CLIENT');
87 |
88 | INSERT INTO FIRMA
89 | VALUES (SEQ_FIRMA.NEXTVAL, 'Profi', 'CLIENT');
90 |
91 | INSERT INTO FIRMA
92 | VALUES (SEQ_FIRMA.NEXTVAL, 'ABOUT YOU', 'CLIENT');
93 |
94 | INSERT INTO FIRMA
95 | VALUES (SEQ_FIRMA.NEXTVAL, 'Deichmann', 'CLIENT');
96 |
97 | INSERT INTO FIRMA
98 | VALUES (SEQ_FIRMA.NEXTVAL, 'H and M', 'CLIENT');
99 |
100 | INSERT INTO FIRMA
101 | VALUES (SEQ_FIRMA.NEXTVAL, 'STOP SEAL GUARD', 'PAZA');
102 |
103 | INSERT INTO FIRMA
104 | VALUES (SEQ_FIRMA.NEXTVAL, 'ATMAN PROTECTION', 'PAZA');
105 |
106 | INSERT INTO FIRMA
107 | VALUES (SEQ_FIRMA.NEXTVAL, 'GARDIENII', 'PAZA');
108 |
109 | INSERT INTO FIRMA
110 | VALUES (SEQ_FIRMA.NEXTVAL, 'QUICK PROTECT', 'PAZA');
111 |
112 | INSERT INTO FIRMA
113 | VALUES (SEQ_FIRMA.NEXTVAL, 'JOHNNY VIP SECURITY', 'PAZA');
114 |
115 | INSERT INTO FIRMA
116 | VALUES (SEQ_FIRMA.NEXTVAL, 'Road Logistics', 'TRANSPORT');
117 |
118 | INSERT INTO FIRMA
119 | VALUES (SEQ_FIRMA.NEXTVAL, 'Transibo', 'TRANSPORT');

```

```

120 |
121 | INSERT INTO FIRMA
122 | VALUES (SEQ_FIRMA.NEXTVAL, 'EasyCargo', 'TRANSPORT');
123 |
124 | INSERT INTO FIRMA
125 | VALUES (SEQ_FIRMA.NEXTVAL, 'Lextom Trans Asd', 'TRANSPORT');
126 |
127 | INSERT INTO FIRMA
128 | VALUES (SEQ_FIRMA.NEXTVAL, 'FedEx', 'TRANSPORT');
129 |
130 | INSERT INTO ECHIPA_PAZA
131 | VALUES (1, 600);
132 |
133 | INSERT INTO ECHIPA_PAZA
134 | VALUES (2, 600);
135 |
136 | INSERT INTO ECHIPA_PAZA
137 | VALUES (3, 800);
138 |
139 | INSERT INTO ECHIPA_PAZA
140 | VALUES (4, 700);
141 |
142 | INSERT INTO ECHIPA_PAZA
143 | VALUES (5, 900);
144 |
145 | INSERT INTO ECHIPA_PAZA
146 | VALUES (6, 1000);
147 |
148 | INSERT INTO ECHIPA_PAZA
149 | VALUES (7, 1000);
150 |
151 | INSERT INTO ECHIPA_PAZA
152 | VALUES (8, 600);
153 |
154 | INSERT INTO ECHIPA_PAZA
155 | VALUES (9, 700);
156 |
157 | INSERT INTO ECHIPA_PAZA
158 | VALUES (10, 600);
159 |
160 | CREATE SEQUENCE SEQ_DEP
161 | INCREMENT BY 500
162 | START WITH 500
163 | MAXVALUE 50000
164 | NOCYCLE;
165 |
166 | INSERT INTO DEPOZIT
167 | VALUES (SEQ_DEP.NEXTVAL, 30, 2, 100);
168 | INSERT INTO DEPOZIT
169 | VALUES (SEQ_DEP.NEXTVAL, 50, 3, 200);
170 | INSERT INTO DEPOZIT
171 | VALUES (SEQ_DEP.NEXTVAL, 80, 1, 300);
172 | INSERT INTO DEPOZIT
173 | VALUES (SEQ_DEP.NEXTVAL, 90, 5, 400);
174 | INSERT INTO DEPOZIT
175 | VALUES (SEQ_DEP.NEXTVAL, 110, 6, 500);
176 | INSERT INTO DEPOZIT
177 | VALUES (SEQ_DEP.NEXTVAL, 100, 4, 1100);
178 | INSERT INTO DEPOZIT
179 | VALUES (SEQ_DEP.NEXTVAL, 120, 10, 1200);
180 | INSERT INTO DEPOZIT
181 | VALUES (SEQ_DEP.NEXTVAL, 140, 8, 1300);

```

```

182 | INSERT INTO DEPOZIT
183 | VALUES (SEQ_DEP.NEXTVAL, 150, 7, 1400);
184 | INSERT INTO DEPOZIT
185 | VALUES (SEQ_DEP.NEXTVAL, 130, 9, 1500);
186 |
187 | CREATE SEQUENCE SEQ_CAM
188 | INCREMENT BY 10
189 | START WITH 15
190 | MAXVALUE 10000
191 | NOCYCLE;
192 |
193 | INSERT INTO CAMION
194 | VALUES (SEQ_CAM.NEXTVAL, 'VOLVO', 'CT12FSD', 1100);
195 |
196 | INSERT INTO CAMION
197 | VALUES (SEQ_CAM.NEXTVAL, 'VOLVO', 'CT90MMM', 1100);
198 |
199 | INSERT INTO CAMION
200 | VALUES (SEQ_CAM.NEXTVAL, 'SCANIA', 'PH13SCN', 1200);
201 |
202 | INSERT INTO CAMION
203 | VALUES (SEQ_CAM.NEXTVAL, 'DAF', 'B55ASD', 1300);
204 |
205 | INSERT INTO CAMION
206 | VALUES (SEQ_CAM.NEXTVAL, 'IVECO', 'MM13IVC', 1400);
207 |
208 | INSERT INTO CAMION
209 | VALUES (SEQ_CAM.NEXTVAL, 'MAN', 'TM20QWE', 1100);
210 |
211 | INSERT INTO CAMION
212 | VALUES (SEQ_CAM.NEXTVAL, 'SCANIA', 'IS32ISA', 1100);
213 |
214 | INSERT INTO CAMION
215 | VALUES (SEQ_CAM.NEXTVAL, 'MAN', 'IS01MAN', 1200);
216 |
217 | INSERT INTO CAMION
218 | VALUES (SEQ_CAM.NEXTVAL, 'DAF', 'TM11IOP', 1300);
219 |
220 | INSERT INTO CAMION
221 | VALUES (SEQ_CAM.NEXTVAL, 'VOLVO', 'MM57RXF', 1500);
222 |
223 | INSERT INTO CAMION
224 | VALUES (SEQ_CAM.NEXTVAL, 'SCANIA', 'B29SCN', 1500);
225 |
226 | CREATE SEQUENCE SEQ_TRAN
227 | INCREMENT BY 10
228 | START WITH 10000
229 | MAXVALUE 2000000000
230 | NOCYCLE;
231 |
232 | INSERT INTO TRANSPORT
233 | VALUES (SEQ_TRAN.NEXTVAL, 3000, 500, 15, to_date('06-05-2021 08:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
234 |
235 | INSERT INTO TRANSPORT
236 | VALUES (SEQ_TRAN.NEXTVAL, 3000, 500, 25, to_date('08-05-2021 06:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
237 |
238 | INSERT INTO TRANSPORT
239 | VALUES (SEQ_TRAN.NEXTVAL, 3500, 1000, 35, to_date('06-05-2021 08:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
240 |

```

```

241 | INSERT INTO TRANSPORT
242 | VALUES (SEQ_TRAN.NEXTVAL, 4000, 1000, 45, to_date('06-05-2021 06:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
243 |
244 | INSERT INTO TRANSPORT
245 | VALUES (SEQ_TRAN.NEXTVAL, 4500, 1000, 55, to_date('06-05-2021 06:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
246 |
247 | INSERT INTO TRANSPORT
248 | VALUES (SEQ_TRAN.NEXTVAL, 5000, 1500, 115, to_date('07-05-2021 08:00', 'dd-mm
    ↳ -yyyy, hh24:mi'));
249 |
250 | INSERT INTO TRANSPORT
251 | VALUES (SEQ_TRAN.NEXTVAL, 5000, 1500, 105, to_date('05-05-2021 06:00', 'dd-mm
    ↳ -yyyy, hh24:mi'));
252 |
253 | INSERT INTO TRANSPORT
254 | VALUES (SEQ_TRAN.NEXTVAL, 3500, 2000, 85, to_date('06-05-2021 06:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
255 |
256 | INSERT INTO TRANSPORT
257 | VALUES (SEQ_TRAN.NEXTVAL, 3500, 2000, 85, to_date('07-05-2021 06:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
258 |
259 | INSERT INTO TRANSPORT
260 | VALUES (SEQ_TRAN.NEXTVAL, 3000, 2500, 25, to_date('10-05-2021 07:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
261 |
262 | INSERT INTO TRANSPORT
263 | VALUES (SEQ_TRAN.NEXTVAL, 3000, 2500, 15, to_date('11-05-2021 05:00', 'dd-mm-
    ↳ yyyy, hh24:mi'));
264 |
265 | CREATE SEQUENCE SEQ_LOTM
266 | INCREMENT BY 5
267 | START WITH 100
268 | MAXVALUE 5000000000
269 | NOCYCLE;
270 |
271 | INSERT INTO LOT_MARFA
272 | VALUES (SEQ_LOTM.NEXTVAL, 'carne miel', 100.00);
273 |
274 | INSERT INTO LOT_MARFA
275 | VALUES (SEQ_LOTM.NEXTVAL, 'carne porc', 300.00);
276 |
277 | INSERT INTO LOT_MARFA
278 | VALUES (SEQ_LOTM.NEXTVAL, 'carne pui', 250.00);
279 |
280 | INSERT INTO LOT_MARFA
281 | VALUES (SEQ_LOTM.NEXTVAL, 'carne pui', 200.00);
282 |
283 | INSERT INTO LOT_MARFA
284 | VALUES (SEQ_LOTM.NEXTVAL, 'rosii', 1000.00);
285 |
286 | INSERT INTO LOT_MARFA
287 | VALUES (SEQ_LOTM.NEXTVAL, 'rosii', 1500.00);
288 |
289 | INSERT INTO LOT_MARFA
290 | VALUES (SEQ_LOTM.NEXTVAL, 'ardei', 500.00);
291 |
292 | INSERT INTO LOT_MARFA
293 | VALUES (SEQ_LOTM.NEXTVAL, 'lapte', 100.00);
294 |

```

```

295 | INSERT INTO LOT_MARFA
296 | VALUES (SEQ_LOTM.NEXTVAL, 'lapte', 100.00);
297 |
298 | INSERT INTO LOT_MARFA
299 | VALUES (SEQ_LOTM.NEXTVAL, 'lapte', 120.00);
300 |
301 | INSERT INTO LOT_MARFA
302 | VALUES (SEQ_LOTM.NEXTVAL, 'tricouri', 100.00);
303 |
304 | INSERT INTO LOT_MARFA
305 | VALUES (SEQ_LOTM.NEXTVAL, 'tricouri', 90.00);
306 |
307 | INSERT INTO LOT_MARFA
308 | VALUES (SEQ_LOTM.NEXTVAL, 'tricouri', 10.00);
309 |
310 | INSERT INTO LOT_MARFA
311 | VALUES (SEQ_LOTM.NEXTVAL, 'pantofi', 20.00);
312 |
313 | INSERT INTO LOT_MARFA
314 | VALUES (SEQ_LOTM.NEXTVAL, 'pantofi', 30.00);
315 |
316 | INSERT INTO LOT_MARFA
317 | VALUES (SEQ_LOTM.NEXTVAL, 'hanorace', 50.00);
318 |
319 | INSERT INTO LOT_MARFA
320 | VALUES (SEQ_LOTM.NEXTVAL, 'hanorace', 35.00);
321 |
322 | INSERT INTO LOT_MARFA
323 | VALUES (SEQ_LOTM.NEXTVAL, 'rochii', 50.00);
324 |
325 | INSERT INTO LOT_MARFA
326 | VALUES (SEQ_LOTM.NEXTVAL, 'fuste', 35.00);
327 |
328 | INSERT INTO LOT_MARFA
329 | VALUES (SEQ_LOTM.NEXTVAL, 'blugi', 35.00);
330 |
331 | CREATE SEQUENCE SEQ_INV_DEP
332 | INCREMENT BY 50
333 | START WITH 100
334 | MAXVALUE 500000000
335 | NOCYCLE;
336 |
337 | --10040
338 | INSERT INTO INVENTAR_DEPOZIT
339 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 100, to_date('02-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
340 | --10040
341 | INSERT INTO INVENTAR_DEPOZIT
342 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 105, to_date('02-04-2021 10:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
343 | --10050
344 | INSERT INTO INVENTAR_DEPOZIT
345 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 110, to_date('01-04-2021 11:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
346 |
347 | INSERT INTO INVENTAR_DEPOZIT
348 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 115, to_date('25-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
349 |
350 | --10060
351 | INSERT INTO INVENTAR_DEPOZIT

```

```

352 | VALUES (SEQ_INV_DEP.NEXTVAL, 3500, 120, to_date('24-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
353 | INSERT INTO INVENTAR_DEPOZIT
354 | VALUES (SEQ_INV_DEP.NEXTVAL, 3500, 130, to_date('07-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
355 |
356 | --10070
357 | INSERT INTO INVENTAR_DEPOZIT
358 | VALUES (SEQ_INV_DEP.NEXTVAL, 4000, 140, to_date('12-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
359 |
360 | --10080
361 | INSERT INTO INVENTAR_DEPOZIT
362 | VALUES (SEQ_INV_DEP.NEXTVAL, 4500, 125, to_date('21-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
363 | INSERT INTO INVENTAR_DEPOZIT
364 | VALUES (SEQ_INV_DEP.NEXTVAL, 4500, 145, to_date('13-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
365 |
366 | --10090
367 | INSERT INTO INVENTAR_DEPOZIT
368 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 195, to_date('11-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
369 |
370 | --10100
371 | INSERT INTO INVENTAR_DEPOZIT
372 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 150, to_date('14-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
373 | INSERT INTO INVENTAR_DEPOZIT
374 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 175, to_date('02-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
375 |
376 | --10110
377 | INSERT INTO INVENTAR_DEPOZIT
378 | VALUES (SEQ_INV_DEP.NEXTVAL, 3500, 165, to_date('03-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
379 |
380 | --10120
381 | INSERT INTO INVENTAR_DEPOZIT
382 | VALUES (SEQ_INV_DEP.NEXTVAL, 3500, 170, to_date('02-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
383 |
384 | --10130
385 | INSERT INTO INVENTAR_DEPOZIT
386 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 190, to_date('06-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
387 | INSERT INTO INVENTAR_DEPOZIT
388 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 185, to_date('01-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
389 |
390 | --10140
391 | INSERT INTO INVENTAR_DEPOZIT
392 | VALUES (SEQ_INV_DEP.NEXTVAL, 3000, 180, to_date('15-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
393 |
394 | INSERT INTO INVENTAR_DEPOZIT
395 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 160, to_date('11-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
396 | INSERT INTO INVENTAR_DEPOZIT
397 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 155, to_date('11-04-2021 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
398 | INSERT INTO INVENTAR_DEPOZIT

```

```

399 | VALUES (SEQ_INV_DEP.NEXTVAL, 5000, 135, to_date('11-04-2021 12:00', 'dd-mm-
    |      ↪ yyyy hh24:mi'), null);
400 |
401 |
402 | CREATE SEQUENCE SEQ_INV_TRAN
403 | INCREMENT BY 55
404 | START WITH 100
405 | MAXVALUE 500000000
406 | NOCYCLE;
407 |
408 | INSERT INTO INVENTAR_TRANSPORT
409 | VALUES (SEQ_INV_TRAN.NEXTVAL, 100, 10000);
410 | INSERT INTO INVENTAR_TRANSPORT
411 | VALUES (SEQ_INV_TRAN.NEXTVAL, 105, 10000);
412 |
413 | INSERT INTO INVENTAR_TRANSPORT
414 | VALUES (SEQ_INV_TRAN.NEXTVAL, 110, 10010);
415 |
416 | INSERT INTO INVENTAR_TRANSPORT
417 | VALUES (SEQ_INV_TRAN.NEXTVAL, 120, 10020);
418 | INSERT INTO INVENTAR_TRANSPORT
419 | VALUES (SEQ_INV_TRAN.NEXTVAL, 130, 10020);
420 |
421 | INSERT INTO INVENTAR_TRANSPORT
422 | VALUES (SEQ_INV_TRAN.NEXTVAL, 140, 10030);
423 |
424 | INSERT INTO INVENTAR_TRANSPORT
425 | VALUES (SEQ_INV_TRAN.NEXTVAL, 125, 10040);
426 | INSERT INTO INVENTAR_TRANSPORT
427 | VALUES (SEQ_INV_TRAN.NEXTVAL, 145, 10040);
428 |
429 | INSERT INTO INVENTAR_TRANSPORT
430 | VALUES (SEQ_INV_TRAN.NEXTVAL, 195, 10050);
431 |
432 | INSERT INTO INVENTAR_TRANSPORT
433 | VALUES (SEQ_INV_TRAN.NEXTVAL, 150, 10060);
434 | INSERT INTO INVENTAR_TRANSPORT
435 | VALUES (SEQ_INV_TRAN.NEXTVAL, 175, 10060);
436 |
437 | INSERT INTO INVENTAR_TRANSPORT
438 | VALUES (SEQ_INV_TRAN.NEXTVAL, 165, 10070);
439 |
440 | INSERT INTO INVENTAR_TRANSPORT
441 | VALUES (SEQ_INV_TRAN.NEXTVAL, 170, 10080);
442 |
443 | INSERT INTO INVENTAR_TRANSPORT
444 | VALUES (SEQ_INV_TRAN.NEXTVAL, 190, 10090);
445 | INSERT INTO INVENTAR_TRANSPORT
446 | VALUES (SEQ_INV_TRAN.NEXTVAL, 185, 10090);
447 |
448 | INSERT INTO INVENTAR_TRANSPORT
449 | VALUES (SEQ_INV_TRAN.NEXTVAL, 180, 10100);
450 |
451 | CREATE SEQUENCE SEQ_ANG
452 | INCREMENT BY 20
453 | START WITH 20000
454 | MAXVALUE 2000000000
455 | NOCYCLE;
456 |
457 | INSERT INTO ANGAJAT
458 | VALUES (SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',

```



```

459 |         to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER',
      |         ↪ null);
460 |
461 | INSERT INTO ANGAJAT
462 | VALUES (SEQ_ANG.NEXTVAL, 'Ovidiu', 'Ion', '0727124356',
463 |         to_date('12-05-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4000, 1100, 'SOFER',
      |         ↪ null);
464 |
465 | INSERT INTO ANGAJAT
466 | VALUES (SEQ_ANG.NEXTVAL, 'Popescu', 'Andrei', '0723323465',
467 |         to_date('11-03-2020 12:00', 'dd-mm-yyyy hh24:mi'), 6000, 1200, 'SOFER',
      |         ↪ null);
468 |
469 | INSERT INTO ANGAJAT
470 | VALUES (SEQ_ANG.NEXTVAL, 'Marian', 'Gheorghe', '0721213456',
471 |         to_date('11-11-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3000, 1200, 'SOFER',
      |         ↪ null);
472 |
473 | INSERT INTO ANGAJAT
474 | VALUES (SEQ_ANG.NEXTVAL, 'Manea', 'George', '0721123959',
475 |         to_date('03-03-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4000, 1300, 'SOFER',
      |         ↪ null);
476 |
477 | INSERT INTO ANGAJAT
478 | VALUES (SEQ_ANG.NEXTVAL, 'Constantin', 'Gheorghe', '0735123456',
479 |         to_date('09-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4500, 1400, 'SOFER',
      |         ↪ null);
480 |
481 | INSERT INTO ANGAJAT
482 | VALUES (SEQ_ANG.NEXTVAL, 'Ion', 'Catalin', '0721126566',
483 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3400, 1500, 'SOFER',
      |         ↪ null);
484 |
485 | INSERT INTO ANGAJAT
486 | VALUES (SEQ_ANG.NEXTVAL, 'Marius', 'Catalin', '072119356',
487 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3300, 600, 'PAZNIC',
      |         ↪ 1);
488 | INSERT INTO ANGAJAT
489 | VALUES (SEQ_ANG.NEXTVAL, 'Grant', 'Ion', '0723123456',
490 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3100, 600, 'PAZNIC',
      |         ↪ 1);
491 |
492 | INSERT INTO ANGAJAT
493 | VALUES (SEQ_ANG.NEXTVAL, 'Ivan', 'George', '0726100456',
494 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3250, 600, 'PAZNIC',
      |         ↪ 2);
495 |
496 | INSERT INTO ANGAJAT
497 | VALUES (SEQ_ANG.NEXTVAL, 'Neagu', 'Ion', '0728123456',
498 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3500, 800, 'PAZNIC',
      |         ↪ 3);
499 |
500 | INSERT INTO ANGAJAT
501 | VALUES (SEQ_ANG.NEXTVAL, 'Popescu', 'Cristian', '0724523456',
502 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3600, 700, 'PAZNIC',
      |         ↪ 4);
503 |
504 | INSERT INTO ANGAJAT
505 | VALUES (SEQ_ANG.NEXTVAL, 'Marius', 'Mihai', '0721223456',
506 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3450, 900, 'PAZNIC',
      |         ↪ 5);
507 |

```

```

508 | INSERT INTO ANGAJAT
509 | VALUES (SEQ_ANG.NEXTVAL, 'Radu', 'Gheorghe', '0724323456',
510 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3375, 1000, 'PAZNIC',
511 |         ↪ 6);
512 | INSERT INTO ANGAJAT
513 | VALUES (SEQ_ANG.NEXTVAL, 'Marin', 'Catalin', '0725623456',
514 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3150, 1000, 'PAZNIC',
515 |         ↪ 7);
516 | INSERT INTO ANGAJAT
517 | VALUES (SEQ_ANG.NEXTVAL, 'Andrei', 'David', '0722123456',
518 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3250, 600, 'PAZNIC',
519 |         ↪ 8);
520 | INSERT INTO ANGAJAT
521 | VALUES (SEQ_ANG.NEXTVAL, 'Popescu', 'Stefan', '0721145456',
522 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3400, 700, 'PAZNIC',
523 |         ↪ 9);
524 | INSERT INTO ANGAJAT
525 | VALUES (SEQ_ANG.NEXTVAL, 'Ion', 'Catalin', '0721765456',
526 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3320, 600, 'PAZNIC',
527 |         ↪ 10);
528 | CREATE SEQUENCE SEQ_IST_CAM
529 | INCREMENT BY 5
530 | START WITH 5
531 | MAXVALUE 2000000000
532 | NOCYCLE;
533 |
534 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
535 | VALUES (SEQ_IST_CAM.NEXTVAL, 20000, 65, to_date('11-04-2020 12:00', 'dd-mm-
536 |         ↪ yyyy hh24:mi'),
537 |         to_date('13-07-2020 12:00', 'dd-mm-yyyy hh24:mi'));
538 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
539 | VALUES (SEQ_IST_CAM.NEXTVAL, 20000, 75, to_date('13-07-2020 12:00', 'dd-mm-
540 |         ↪ yyyy hh24:mi'),
541 |         to_date('21-11-2020 12:00', 'dd-mm-yyyy hh24:mi'));
542 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
543 | VALUES (SEQ_IST_CAM.NEXTVAL, 20000, 15, to_date('21-11-2020 12:00', 'dd-mm-
544 |         ↪ yyyy hh24:mi'), null);
545 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
546 | VALUES (SEQ_IST_CAM.NEXTVAL, 20020, 25, to_date('12-05-2020 12:00', 'dd-mm-
547 |         ↪ yyyy hh24:mi'), null);
548 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
549 | VALUES (SEQ_IST_CAM.NEXTVAL, 20040, 85, to_date('11-03-2020 12:00', 'dd-mm-
550 |         ↪ yyyy hh24:mi'), null);
551 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
552 | VALUES (SEQ_IST_CAM.NEXTVAL, 20060, 35, to_date('11-11-2020 12:00', 'dd-mm-
553 |         ↪ yyyy hh24:mi'), null);
554 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
555 | VALUES (SEQ_IST_CAM.NEXTVAL, 20080, 45, to_date('03-03-2020 12:00', 'dd-mm-
556 |         ↪ yyyy hh24:mi'), null);
557 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE

```

```

558 | VALUES (SEQ_IST_CAM.NEXTVAL, 20100, 55, to_date('09-04-2020 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
559 |
560 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
561 | VALUES (SEQ_IST_CAM.NEXTVAL, 20120, 105, to_date('02-07-2020 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'),
562 |         to_date('06-06-2020 12:00', 'dd-mm-yyyy hh24:mi'));
563 |
564 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
565 | VALUES (SEQ_IST_CAM.NEXTVAL, 20120, 115, to_date('06-06-2020 12:00', 'dd-mm-
      ↳ yyyy hh24:mi'), null);
566 |
567 | CREATE SEQUENCE SEQ_PERM
568 | INCREMENT BY 2
569 | START WITH 1
570 | MAXVALUE 10000000000
571 | NOCYCLE;
572 |
573 | INSERT INTO PERMIS
574 | VALUES (SEQ_PERM.NEXTVAL, 20000, 'C', to_date('06-06-2017 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
575 |
576 | INSERT INTO PERMIS
577 | VALUES (SEQ_PERM.NEXTVAL, 20000, 'A', to_date('06-06-2013 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
578 |
579 | INSERT INTO PERMIS
580 | VALUES (SEQ_PERM.NEXTVAL, 20000, 'B', to_date('06-06-2012 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
581 |
582 | INSERT INTO PERMIS
583 | VALUES (SEQ_PERM.NEXTVAL, 20020, 'C', to_date('06-06-2018 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
584 |
585 | INSERT INTO PERMIS
586 | VALUES (SEQ_PERM.NEXTVAL, 20040, 'C', to_date('06-05-2016 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
587 |
588 | INSERT INTO PERMIS
589 | VALUES (SEQ_PERM.NEXTVAL, 20060, 'C', to_date('03-11-2014 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
590 |
591 | INSERT INTO PERMIS
592 | VALUES (SEQ_PERM.NEXTVAL, 20080, 'C', to_date('03-03-2002 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
593 |
594 | INSERT INTO PERMIS
595 | VALUES (SEQ_PERM.NEXTVAL, 20100, 'B', to_date('06-11-2010 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
596 |
597 | INSERT INTO PERMIS
598 | VALUES (SEQ_PERM.NEXTVAL, 20100, 'C', to_date('06-11-2012 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
599 |
600 | INSERT INTO PERMIS
601 | VALUES (SEQ_PERM.NEXTVAL, 20120, 'C', to_date('12-06-2014 12:00', 'dd-mm-yyyy
      ↳ hh24:mi'));
602 |
603 | CREATE SEQUENCE SEQ_IST_POP
604 | INCREMENT BY 5
605 | START WITH 50
606 | MAXVALUE 10000000000000000000

```

```

607 | NOCYCLE;
608 |
609 | INSERT INTO ISTORIC_POPASURI
610 | VALUES (SEQ_IST_POP.NEXTVAL, 180, 20000, to_date('06-05-2021 10:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 12:00', 'dd-mm-yyyy hh24:mi'));
611 |
612 | INSERT INTO ISTORIC_POPASURI
613 | VALUES (SEQ_IST_POP.NEXTVAL, 120, 20000, to_date('06-05-2021 21:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 21:30', 'dd-mm-yyyy hh24:mi'));
614 |
615 | INSERT INTO ISTORIC_POPASURI
616 | VALUES (SEQ_IST_POP.NEXTVAL, 180, 20020, to_date('08-05-2021 10:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('08-05-2021 12:00', 'dd-mm-yyyy hh24:mi'));
617 |
618 | INSERT INTO ISTORIC_POPASURI
619 | VALUES (SEQ_IST_POP.NEXTVAL, 120, 20020, to_date('08-05-2021 21:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('08-05-2021 21:30', 'dd-mm-yyyy hh24:mi'));
620 |
621 | INSERT INTO ISTORIC_POPASURI
622 | VALUES (SEQ_IST_POP.NEXTVAL, 100, 20060, to_date('06-05-2021 10:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 10:20', 'dd-mm-yyyy hh24:mi'));
623 |
624 | INSERT INTO ISTORIC_POPASURI
625 | VALUES (SEQ_IST_POP.NEXTVAL, 140, 20060, to_date('06-05-2021 19:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 19:30', 'dd-mm-yyyy hh24:mi'));
626 |
627 | INSERT INTO ISTORIC_POPASURI
628 | VALUES (SEQ_IST_POP.NEXTVAL, 140, 20080, to_date('06-05-2021 17:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
629 |
630 | INSERT INTO ISTORIC_POPASURI
631 | VALUES (SEQ_IST_POP.NEXTVAL, 140, 20100, to_date('06-05-2021 17:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('06-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
632 |
633 | INSERT INTO ISTORIC_POPASURI
634 | VALUES (SEQ_IST_POP.NEXTVAL, 180, 20120, to_date('07-05-2021 17:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('07-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
635 |
636 | INSERT INTO ISTORIC_POPASURI
637 | VALUES (SEQ_IST_POP.NEXTVAL, 180, 20120, to_date('05-05-2021 17:00', 'dd-mm-
        ↳ yyyy hh24:mi'), to_date('05-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
638 |
639 | commit; -- salvam inserarile

```

14 Cereri SQL - ex11

```
01 | /* 11. Formulati in limbaj natural si implementati 5 cereri SQL complexe ce
    | ↪ vor
02 | * utiliza, in ansamblul lor, urmatoarele elemente:
03 | * - operatie join pe cel putin 4 tabele
04 | * - filtrare la nivel de linii
05 | * - subcereri sincronizate in care intervin cel putin 3 tabele
06 | * - subcereri nesincronizate in care intervin cel putin 3 tabele
07 | * - grupari de date, functii grup, filtrare la nivel de grupuri
08 | * - ordonari
09 | * - utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe
    | ↪ date
10 | * calendaristice, a functiilor NVL si DECODE, a cel putin unei expresii
    | ↪ CASE
11 | * utilizarea a cel putin 1 bloc de cerere (clauza WITH) */
12 |
13 | -- 1
14 |
15 | /* Afisati pentru toti soferi toate popasurile in care s-au oprit in ziua in
    | ↪ care
16 | * s-a oprit pentru prima data un sofer la restaurantul Ceptura din Ploiesti
    | ↪ , sa
17 | * se afiseze daca soferul a venit cu un camion inmatriculat in Prahova sau
    | ↪ in
18 | * alta regiune, numele, prenumele si salariul soferului, numele si tipul
19 | * popasului la care s-a oprit, la ce ora a sosit, cat a stat, cati bani a
20 | * cheltuit la popas, si locatia popasului. Locatia sa fie afisata pe o
    | ↪ singura
21 | * coloana numita Locatie popas. Un sofer cheltuiește la un popas in functie
    | ↪ de
22 | * salariul sau, soferii cu un salariu mai mic de 3000lei inclusiv vor
    | ↪ cheltui doar
23 | * 0.005% din salariu, cei cu salariu intre (3000, 4500] vor cheltui 0.01%
    | ↪ din
24 | * salariu, iar restul vor cheltui 0.015% din salariu. Sa se ordoneze dupa
    | ↪ nume
25 | * si dupa prenume. Sa se noteze restul coloanelor corespunzator */
26 |
27 | WITH psp(ds) AS -- data primei opriri a unui sofer la Restaurantul Ceptura
    | ↪ din Ploiesti
28 |     (SELECT to_char(min(sip.data_sosire), 'dd-mm-yyyy')
29 |      FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie)
30 |      JOIN istoric_popasuri sip ON (sp.id_popas = sip.id_popas
    | ↪ )
31 |      WHERE initcap(sp.numa) = 'Restaurant Ceptura' and initcap(sl.localitate
    | ↪ ) = 'Ploiesti'
32 |     )
33 |
34 | SELECT DECODE((SELECT upper(substr(sc.nr_inmatriculare, 0, 2))
35 |               FROM camion sc JOIN istoric_camioane_conduse sic ON (sc.
    | ↪ id_camion = sic.id_camion)
36 |               JOIN angajat sa ON (sic.id_angajat = sa.
    | ↪ id_angajat)
37 |               WHERE sic.id_angajat = a.id_angajat
38 |                 and ip.data_sosire < NVL(sic.data_sfarsit, sysdate)
39 |                 and ip.data_sosire > NVL(sic.data_inceput, sysdate)
40 |                 ), 'PH', 'Din Prahova', 'Alta regiune'
41 |                ) "Verifica camion din Prahova",
42 |      a.numa "Nume angajat", a.prenume "Prenume angajat", a.salariu "
    | ↪ Salariu",
43 |      p.numa "Nume popas", initcap(p.tip_popas) "Tip popas",
```

```

44 |         to_char(ip.data_sosire, 'hh24:mi') "Ora sosire",
45 |         substr(numtodsinterval((ip.data_plecare - ip.data_sosire), 'DAY'),
    |         ↪ 12, 5) "Durata",
46 |         (SELECT CASE WHEN salariu <=3000 THEN salariu * 0.005
47 |                WHEN salariu <=4500 THEN salariu * 0.01
48 |                WHEN salariu > 4500 THEN salariu * 0.015
49 |         END
50 |         FROM angajat ssa
51 |         WHERE a.id_angajat = ssa.id_angajat
52 |         ) || 'lei' "bani cheltuiti",
53 |         'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-') ||
54 |         ' Str. ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-')
    |         ↪ "Locatie popas"
55 | FROM istoric_popasuri ip JOIN angajat a ON(ip.id_angajat = a.id_angajat)
56 |                        JOIN popas p ON(ip.id_popas = p.id_popas)
57 |                        JOIN locatie l ON(p.id_locatie = l.id_locatie)
58 | WHERE a.tip_angajat = 'SOFER'
59 |       and to_char(data_sosire, 'dd-mm-yyyy') = (SELECT * FROM psp)
60 | ORDER BY a.numa, a.prenume;
61 |
62 | -- 2
63 |
64 | /* Sa se afiseze numele, prenumele, ce salariu isi doreste, cat de bine este
65 | * platit, vechimea la locul de munca curent, la ce firma lucreaza, si ce
66 | * depozit pazesc toti paznicii care lucreaza in depozite din localitatea in
    |     ↪ care
67 | * se afla cel mai vizitat popas de soferii de tir si doar pentru paznicii
    |     ↪ pentru
68 | * care a intrat sub paza lor in depozit cel putin un transport de marfa.
69 | * Daca un paznic are salariul mai mic decat media salariul din firma in
    |     ↪ care
70 | * lucreaza, atunci el isi doreste o marire de salariu cu 200 lei, in caz
    |     ↪ contra
71 | * doresc o marire de 100 lei. Consideram ca un paznic este bine platit daca
    |     ↪ are
72 | * un salariu mai mare decat 3000 lei inclusiv. Suma de 3000lei reprezinta
    |     ↪ media
73 | * salariilor paznicilor din Romania inidiferent de locatia pazita(nu intra
    |     ↪ doar
74 | * depozitele din gestiunea acestei baze de date). Sa se odroneze
    |     ↪ descrescator
75 | * dupa salariu si in caz de egalitate crescator dupa nume si prenume. */
76 |
77 | WITH
78 |     avgs AS -- media salariilor paznicilor per firma
79 |     (SELECT round(avg(salariu)) salariu, id_firma
80 |      FROM angajat
81 |      WHERE tip_angajat = 'PAZNIC'
82 |      GROUP BY id_firma
83 |     ),
84 |
85 |     cm_vizitat_popas AS -- id-ul celui mai vizitat popas
86 |     (
87 |         SELECT id_popas popas
88 |         FROM istoric_popasuri
89 |         GROUP BY id_popas
90 |         HAVING count(id_popas) = (SELECT max(count(id_popas))
91 |                                FROM istoric_popasuri
92 |                                GROUP BY id_popas
93 |                                )
94 |     ),
95 |

```

```

196 | loc_cm_vizitat_popas AS -- localitatea in care se afla cel mai vizitat
197 |   ↳ popas
198 | (SELECT sl.localitate localitate
199 |   FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie),
200 |   cm_vizitat_popas cmvp
201 |   WHERE sp.id_popas = cmvp.popas
202 |   )
203 |
204 | SELECT a.numa "Nume paznic", a.prenume "Prenume paznic",
205 |       CASE WHEN a.salariu < avgs.salariu THEN a.salariu + 200
206 |       ELSE a.salariu + 100
207 |       END "Salariu dorit",
208 |       DECODE((a.salariu-3000)-abs(a.salariu-3000)
209 |         , 0, 'bine platit', 'prost platit') "Cat de bine este platit",
210 |       concat(substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'),
211 |   ↳ 8, 3) || ' zile ',
212 |   ↳ substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'), 12,
213 |   ↳ 5)) "Vechime angajat",
214 |       f.numa "Nume firma",
215 |       'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-')
216 |   ↳ ||
217 |       ' Str. ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-')
218 |   ↳ "Locatie depozit repartizat",
219 |
220 |       ( -- nr de colegi ai unui angajat
221 |       SELECT DECODE(count(id echipa_paza)-1, 0, 'Nu are colegi', 1,
222 |         to_char(count(id echipa_paza)-1) || ' colegi',
223 |         to_char(count(id echipa_paza)-1) || ' colegi')
224 |       FROM angajat
225 |       WHERE id echipa_paza = a.id echipa_paza
226 |       GROUP BY id echipa_paza
227 |       ) "Nr. Colegi"
228 |
229 | FROM echipa_paza ep JOIN firma f ON (ep.id_firma = f.id_firma)
230 |   JOIN angajat a ON (ep.id echipa_paza = a.id echipa_paza)
231 |   JOIN depozit d ON (ep.id echipa_paza = d.id echipa_paza)
232 |   JOIN locatie l ON (d.id_locatie = l.id_locatie)
233 |   JOIN avgs ON (avgs.id_firma = f.id_firma),
234 |   loc_cm_vizitat_popas cmvp
235 |
236 | WHERE a.tip_angajat = 'PAZNIC'
237 |   and l.localitate in cmvp.localitate
238 |   and ( -- cate transporturi de marfa au intrat in depozit sub paza
239 |   ↳ fiecarui angajat
240 |       SELECT count(depozit_destinatie)
241 |       FROM transport
242 |       WHERE depozit_destinatie = d.id_depozit
243 |         and data_plecure > a.data_angajare
244 |       GROUP BY depozit_destinatie
245 |       ) >= 1
246 | ORDER BY a.salariu desc, a.numa, a.prenume;
247 |
248 | -- 3
249 |
250 | /* Sa se afiseze data\&ora de plecure, numele si prenumele soferului, ce
251 |   ↳ salariu
252 | * are, cat de bine este platit fata de media salariului din Ro care este de
253 | * 3500lei, ce marca de camion a condus, nr de inmatriculare al camionului,
254 | * sa se specifice in ce judet a fost inmatriculat camionul,
255 | * iar pentru depozitul de plecure si destinatie sa se precizeze ce firma le
256 |   ↳ detine

```

```

150 | * si in ce zona se afla, pentru fiecare transport al celor mai bine platiti
151 | * soferi din firmele unde sunt angajati, sa se afiseze doar transporturile
152 | * in care soferii au folosit acelasi camion ca cel actual pe care il conduc
    | ↳ si
153 | * doar transporturile realizate incepand cu data in care a fost angajat
    | ↳ ultimul
154 | * sofer de la toate firmele de transport.
155 | * Sa se ordoneze detaliile anterioare despre transport in ordine
    | ↳ crescatoare
156 | * dupa data de plecare din primul depozit */
157 |
158 | WITH ac AS -- informatii despre cei mai bine platiti soferi din firma unde
    | ↳ sunt angajati si camionul curent
159 | (SELECT c.id_camion, c.marca, c.nr_inmatriculare, a.id_angajat, a.numa,
160 |        a.preume, a.salaru, a.id_firma
161 | FROM angajat a JOIN istoric_camioane_conduse ic ON (a.id_angajat = ic.
    | ↳ id_angajat)
162 |        JOIN camion c ON(c.id_camion = ic.id_camion)
163 | WHERE (a.salaru, a.id_firma) in (SELECT max(salaru), id_firma --
    | ↳ obtine salariul mare din firma
164 |                                FROM angajat
165 |                                WHERE tip_angajat = 'SOFER'
166 |                                GROUP BY tip_angajat, id_firma
167 |                                )
168 | and ic.data_sfarsit is null
169 | )
170 |
171 | SELECT
172 |     to_char(t.data_plecara, 'dd-mm-yyyy') "Data plecara",
173 |     to_char(t.data_plecara, 'hh24:mi') "Ora",-- data plecara transport
174 |     ac.numa "Nume sofer", ac.preume "Preume", -- cine a condus acest
    | ↳ transport
175 |     ac.salaru "Salaru sofer", -- ce salaru are
176 |     DECODE((ac.salaru-3500)-abs(ac.salaru-3500)
177 |     , 0, 'bine platit', 'prost platit') "Cat de bine este platit",
178 |
179 |     ac.marca "Camion",
180 |     ac.nr_inmatriculare "Nr inmatriculare",
181 |     CASE WHEN substr(ac.nr_inmatriculare, 0, 2)= 'MM' THEN 'Maramures'
182 |           WHEN substr(ac.nr_inmatriculare, 0, 2)= 'IS' THEN 'Iasi'
183 |           WHEN substr(ac.nr_inmatriculare, 0, 2)= 'CT' THEN 'Constanta'
184 |           WHEN substr(ac.nr_inmatriculare, 0, 1)= 'B' THEN 'Bucuresti'
185 |           ELSE 'Necunoscut'
186 |     END "Inmatriculat in judetul",
187 |
188 |     (SELECT sf.numa -- cine detine depozitul de plecara
189 |     FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
190 |     WHERE sd.id_depozit = t.depozit_plecara) "Proprietar depozit plecara",
191 |     (SELECT 'Loc: ' || NVL(sl.localitate, '-') || ' Str. ' -- unde se afla
    | ↳ depozitul
192 |         || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-')
193 |     FROM depozit sd JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
194 |     WHERE sd.id_depozit = t.depozit_plecara) "Locatie depozit plecara",
195 |
196 |     (SELECT sf.numa -- cine detine depozitul de destinatie
197 |     FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
198 |     WHERE sd.id_depozit = t.depozit_destinatie) "Proprietar depozit
    | ↳ destinatie",
199 |     (SELECT 'Loc: ' || NVL(sl.localitate, '-') || ' Str. ' -- unde se afla
    | ↳ depozitul
200 |         || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-')
201 |     FROM depozit sd JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)

```



```

202 |         WHERE sd.id_depozit = t.depozit_destinatie) "Locatie depozit destinatie
      |         ↪ "
203 |
204 | FROM transport t JOIN ac ON(t.id_camion = ac.id_camion)
205 |
206 | WHERE t.data_plecure > (SELECT max(data_angajare)
207 |                        FROM angajat
208 |                        WHERE tip_angajat = 'SOFER')
209 | ORDER BY t.data_plecure;
210 |
211 | -- 4
212 |
213 | /* Sa se afiseze pentru transporturile realizate dupa prima zi de luni de la
      |     ↪ data
214 | * primei angajari a unui sofer in firma care detine camionul care a
      |     ↪ realizat
215 | * transportul, numele marfii, cantitatea de marfa, tipul marfii
      |     ↪ transportate
216 | * (putem avea alimente, imbracaminte, incaltaminte), marca, nr de
      |     ↪ inmatriculare
217 | * si judetul in care a fost inmatriculat camionul cu care a fost realizat
218 | * transportul, data si ora la care a plecat camionul din depozitul de
      |     ↪ plecare,
219 | * ce firma detine depozitul de plecare, in ce locatie se afla depozitul de
220 | * plecare, anaog pentru cel de destinatie. Sa se ordoneze crescator dupa
      |     ↪ data
221 | * si ora de plecare, si descrescator dupa cantitatea de marfa transportata
      |     ↪ */
222 |
223 | WITH depozit_plecure AS
224 |     (SELECT id_depozit, sf.numa detinator, 'Loc: ' || NVL(sl.localitate, '-')
      |     ↪ )
225 |     || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr),
      |     ↪ '-') locatie
226 |     FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
227 |     JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
228 |     ),
229 |     depozit_destinatie AS
230 |     (SELECT id_depozit, sf.numa detinator, 'Loc: ' || NVL(sl.localitate, '-')
      |     ↪ )
231 |     || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr),
      |     ↪ '-') locatie
232 |     FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
233 |     JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
234 |     )
235 |
236 | SELECT lm.numa "Marfa", lm.cantitate "Cantitate marfa",
237 |        DECODE(lower(lm.numa), 'carne miel', 'alimente', 'carne porc', '
      |     ↪ alimente',
238 |        'carne pui', 'alimente', 'lapte', 'alimente', 'rosii', '
      |     ↪ alimente',
239 |        'ardei', 'alimente', 'tricouri', 'imbracaminte', 'hanorace', '
      |     ↪ imbracaminte',
240 |        'rochii', 'imbracaminte', 'fuste', 'imbracaminte', 'blugi', '
      |     ↪ imbracaminte',
241 |        'pantofi', 'incaltaminte', 'necunoscut') "Tip aliment",
242 |        c.marca "Marca camion", c.nr_inmatriculare "Nr. inmatriculare",
243 |        CASE WHEN substr(c.nr_inmatriculare, 0, 2)= 'MM' THEN 'Maramures'
244 |              WHEN substr(c.nr_inmatriculare, 0, 2)= 'IS' THEN 'Iasi'
245 |              WHEN substr(c.nr_inmatriculare, 0, 2)= 'CT' THEN 'Constanta'
246 |              WHEN substr(c.nr_inmatriculare, 0, 1)= 'B' THEN 'Bucuresti'
247 |              ELSE 'Necunoscut'

```

```

248 |         END "Inmatriculat in judetul",
249 |         to_char(t.data_plecure, 'dd-mm-yyyy hh24:mi') "Data si ora plecare",
250 |         round(months_between(sysdate, t.data_plecure)) "Nr. luni de la
    |         ↳ transport",
251 |         dp.detinator "Detinator depozit plecare", dp.locatie "Locatie depozit
    |         ↳ plecare",
252 |         dd.detinator "Detinator depozit destinatie", dd.locatie "Locatie
    |         ↳ depozit destinatie"
253 |
254 | FROM transport t JOIN camion c ON(t.id_camion = c.id_camion)
255 |                 JOIN firma f ON(c.id_firma = f.id_firma)
256 |                 JOIN inventar_transport it ON(t.id_transport = it.
    |                 ↳ id_transport)
257 |                 JOIN lot_marfa lm ON(it.id_lot_marfa = lm.id_lot_marfa)
258 |                 JOIN depozit_plecure dp ON(dp.id_depozit = t.
    |                 ↳ depozit_plecure)
259 |                 JOIN depozit_destinatie dd ON(dd.id_depozit = t.
    |                 ↳ depozit_destinatie)
260 |
261 |
262 | WHERE data_plecure >= next_day((SELECT max(sa.data_angajare)
    |                               FROM angajat sa JOIN
    |                               ↳ istoric_camioane_conduse sic ON(sa.id_angajat = sic.id_angajat)
263 |                               JOIN camion sc ON(sic.
    |                               ↳ id_camion = sc.id_camion)
264 |                               WHERE sa.tip_angajat = 'SOFER' and sa.
    |                               ↳ id_firma = f.id_firma
265 |                               GROUP BY tip_angajat
266 |                               ), 'MONDAY')
267 |
268 |
269 | ORDER BY t.data_plecure, lm.cantitate desc;
270 |
271 | -- 5
272 |
273 | /* Pentru toti soferii care lucreaza la una din firmele Road Logistics,
    |     ↳ Transibo
274 | * sau Lextom Trans Asd si care detin cel putin 2 categorii de permis sa se
275 | * afiseze numele&prenumele soferului, cat de experimentat e(<1an e
    |     ↳ INCEPATOR,
276 | * <700 zile AVANSAT si in rest PROFESIONIST), cat de bine este platit(se
    |     ↳ considera
277 | * ca un sofer este bine platit daca are un salariu>=4000lei) marca
    |     ↳ camionului
278 | * si nr de inmatriculare al camionului pe care l-a condus in acel transport
    |     ↳ ,
279 | * data in care a plecat camionul din depozit cu marfa, detinatorul
    |     ↳ depozitului
280 | * de plecare si destinatie, si analog pentru cel de destinatie */
281 |
282 | WITH soferi AS -- soferii care detin cel putin 2 categorii de permis auto si
    |     ↳ care
283 |     -- lucreaza la una din firmele: Road Logistics, Transibo, Lextom Trans
    |     ↳ Asd
284 |     (SELECT a.id_angajat id_sofer
285 |      FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
286 |      JOIN firma f ON (f.id_firma = a.id_firma)
287 |      WHERE initcap(f.num) = 'Road Logistics' or initcap(f.num) = 'Transibo
    |      ↳ '
288 |      or initcap(f.num) = 'Lextom Trans Asd'
289 |      GROUP BY a.id_angajat
290 |      HAVING count(p.id_angajat) >= 2
291 |      )

```

```

292 |
293 | SELECT a.num "Nume sofer", a.prenume "Prenume",
294 |      CASE WHEN to_number(substr(numtodsinterval((sysdate - a.data_angajare
    ↳ ), 'DAY'), 2, 9)) <= 356 THEN 'INCEPATOR'
295 |           WHEN to_number(substr(numtodsinterval((sysdate - a.data_angajare
    ↳ ), 'DAY'), 2, 9)) <= 700 THEN 'AVANSAT'
296 |           ELSE 'PROFESIONIST'
297 |      END "Experienta",
298 |      DECODE((a.salariu-4000)-abs(a.salariu-4000), 0, 'bine platit', 'prost
    ↳ platit') "Cat de bine este platit",
299 |      c.marca "Marca camion", c.nr_inmatriculare "Nr. inmatriculare",
300 |      (SELECT sf.num -- cine detine depozitul de plecare
301 |       FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
302 |       WHERE sd.id_depozit = t.depozit_plecare) "Proprietar depozit plecare
    ↳ ",
303 |      (SELECT 'Loc: ' || NVL(sl.localitate, '-') || ' Str. ' -- unde se
    ↳ afla depozitul
304 |           || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-')
    ↳ )
305 |      FROM depozit sd JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
306 |      WHERE sd.id_depozit = t.depozit_plecare) "Locatie depozit plecare",
307 |
308 |      (SELECT sf.num -- cine detine depozitul de destinatie
309 |       FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
310 |       WHERE sd.id_depozit = t.depozit_destinatie) "Proprietar depozit
    ↳ destinatie",
311 |      (SELECT 'Loc: ' || NVL(sl.localitate, '-') || ' Str. ' -- unde se
    ↳ afla depozitul
312 |           || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-')
    ↳ )
313 |      FROM depozit sd JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
314 |      WHERE sd.id_depozit = t.depozit_destinatie) "Locatie depozit
    ↳ destinatie"
315 |
316 | FROM angajat a JOIN istoric_camioane_conduse ic ON(a.id_angajat = ic.
    ↳ id_angajat)
317 |      JOIN camion c ON(ic.id_camion = c.id_camion)
318 |      JOIN transport t ON(c.id_camion = t.id_camion)
319 | WHERE a.id_angajat in (SELECT * FROM soferi)
320 |      and ic.data_sfarsit is null;

```

15 Operatii de actualizare si suprimare a datelor - ex12

```
01 | /* 12. Implementarea a 3 operatii de actualizare sau suprimare a datelor
02 | * utilizand subcereri. */
03 |
04 | /* pentru inceputul verii fiecare firma de paza ofera o marire de salariu cu
    ↳ 75lei
05 | * angajatilor cu cel mai mic salariu din firma */
06 | UPDATE ANGAJAT
07 | SET salariu = salariu + 75
08 | WHERE (salariu, id_firma) in (SELECT min(salariu), id_firma
09 |                               FROM ANGAJAT
10 |                               WHERE tip_angajat = 'PAZNIC'
11 |                               GROUP BY id_firma
12 |                               );
13 |
14 | /* pe data de 2 aprilie 2021 a intrat in inventarierea unui depozit din
15 | * jud Tulcea loc Tulcea str. Unirii nr. 213 un lot de marfa
16 | * de carne de miel a carei cantitate a fost introdusa gresit in baza de
    ↳ date,
17 | * in loc 400 de kg de carne a fost introdusa valoarea 100 */
18 | UPDATE LOT_MARFA
19 | SET cantitate = 400
20 | WHERE lower(nume) = 'carne miel'
21 |       AND id_lot_marfa IN
22 |       (SELECT id_lot_marfa
23 |        FROM LOCATIE l JOIN DEPOZIT d ON(l.id_locatie = d.id_locatie)
24 |        JOIN INVENTAR_DEPOZIT id ON(d.id_depozit = id.
    ↳ id_depozit)
25 |        WHERE initcap(localitate) = 'Tulcea' AND initcap(strada)='Unirii'
    ↳ AND nr = 213
26 |        AND to_char(id.data_sosire, 'dd-mm-yyyy') = '02-04-2021'
27 |       );
28 |
29 | /* stregem din baza de date popasurile care nu au fost vizitate pana acum de
30 | * niciun sofer de tir */
31 | DELETE FROM POPAS
32 | WHERE id_popas not in (select id_popas FROM ISTORIC_POPASURI);
33 |
34 | commit; -- salvam modificarile facute
```

16 Cereri outer-join + division - ex16

```
01 | /* 16. Formulati in limbaj natural si implementati in SQL: o cerere ce
    | ↳ utilizeaza
02 | * operatia outer-join pe minimum 4 tabele si doua cereri ce utilizeaza
    | ↳ operatia
03 | * division. */
04 |
05 | -- outer join - versiunea 1 va fi optimizata la cerinta 17
06 |
07 | /* Pentru fiecare camion sa se afiseze marca, nr. de inmatriculare, numele
    | ↳ firmei,
08 | * de ce soferi a fost condus(pentru fiecare sofer sa se precizeze numele,
09 | * prenumele si salariul, in cazul in care un camion nu a fost condus de un
10 | * sofer sa se inlocuiasca aceste coloane cu '-'), dar si data de plecare
    | ↳ pentru
11 | * fiecare transport realizat de sofer cu acest camion. Daca unui angajat i-
    | ↳ a
12 | * fost alocat un camion dar nu a realizat niciun transport in loc de data
    | ↳ sa
13 | * fie afisat mesajul 'nu a realizat transporturi'. Sa se afiseze
    | ↳ informatiile
14 | * despre angajatii doar daca sunt soferi, au fost angajati inainte de data
    | ↳ de
15 | * 13 sept 2020 si detin cel putin un permis auto, in cazul in care un
    | ↳ camion a
16 | * fost condus de un sofer care nu indeplineste aceste cerinte se vor
    | ↳ inlocui
17 | * coloanele cu informatii despre angajat cu '-'.
18 | * Sa se excluda camioanele de marca IVECO. */
19 |
20 | WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin cel mult
    | ↳ 2 permise auto
21 | (SELECT p.id_angajat, a.numa, a.prenume, a.salariu
22 | FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
23 | WHERE a.tip_angajat = 'SOFER'
24 | and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
25 | GROUP BY p.id_angajat, a.numa, a.prenume, a.salariu
26 | HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1 permis
27 | )
28 |
29 | SELECT c.marca, c.nr_inmatriculare, f.numa,
30 | NVL(a.numa, '-') "Nume sofer", NVL(a.prenume, '-') "Prenume sofer",
31 | NVL(to_char(a.salariu), '-') "Salariu sofer",
32 | NVL(to_char(t.data_plecare, 'dd-mm-yyyy'), 'nu a realizat
    | ↳ transporturi') "Data transport"
33 | FROM camion c JOIN firma f ON (c.id_firma = f.id_firma)
34 | FULL OUTER JOIN istoric_camioane_conduse icc ON (c.id_camion =
    | ↳ icc.id_camion)
35 | FULL OUTER JOIN ang a ON (icc.id_angajat = a.id_angajat)
36 | FULL OUTER JOIN transport t ON (c.id_camion = t.id_camion)
37 | WHERE
38 | upper(c.marca) != 'IVECO' and
39 | f.tip_firma = 'TRANSPORT' and
40 | ((icc.data_inceput < t.data_plecare and t.data_plecare <= NVL(icc.
    | ↳ data_sfarsit, sysdate)
41 | ) or t.data_plecare is null
42 | );
43 |
44 | -- division
45 |
46 | -- 1
```

```

47 |
48 | /* Sa se afiseze pentru fiecare sofer numele, prenumele, salariul si nr. de
49 | * permise auto detinute, care au condus cel mult aceleasi marci de camioane
50 | * precum soferul Ion Vasile. Sa se ordoneze descrescator dupa nr. de
   |   ↳ permise
51 | * detinute, descrescator dupa salariu in caz de egalitate si in final
   |   ↳ crescator
52 | * dupa nume\&prenume */
53 |
54 | WITH marca_camion_ang AS -- marcile de camioane conduse de angajatul Ion
   |   ↳ Vasile
55 |     (select c.marca
56 |      from camion c join istoric_camioane_conduse icc on (c.id_camion =
   |   ↳ icc.id_camion)
57 |      join angajat a on (icc.id_angajat = a.id_angajat)
58 |      where initcap(a.num) = 'Ion' and initcap(a.prenume) = 'Vasile'
59 |      ),
60 |
61 |     division_ang AS -- id-urile angajatilor care conduc cel mult aceleasi
   |   ↳ marci de camioane precum ang Ion Vasile (inclusiv el)
62 |     (select icc.id_angajat id_angajat
63 |      from camion c join istoric_camioane_conduse icc on (c.id_camion =
   |   ↳ icc.id_camion)
64 |      where c.marca in
65 |          (SELECT *
66 |           FROM marca_camion_ang
67 |          )
68 |      group by icc.id_angajat
69 |      having count(*) <=
70 |          (SELECT count(*)
71 |           FROM marca_camion_ang
72 |          )
73 |
74 |     MINUS
75 |
76 |     select icc.id_angajat
77 |     from camion c join istoric_camioane_conduse icc on (c.id_camion =
   |   ↳ icc.id_camion)
78 |     where c.marca not in (SELECT *
79 |                          FROM marca_camion_ang
80 |                          )
81 |     )
82 |
83 | SELECT a.num, a.prenume, a.salariu, count(*) "Nr. permise auto"
84 | FROM angajat a JOIN division_ang da ON (a.id_angajat = da.id_angajat)
85 |      JOIN permis p ON (a.id_angajat = p.id_angajat)
86 | WHERE not (initcap(a.num) = 'Ion' and initcap(a.prenume) = 'Vasile')
87 | GROUP BY a.num, a.prenume, a.salariu
88 | ORDER BY 4 desc, 3 desc, 1, 2;
89 |
90 | -- 2
91 |
92 | /* Sa se afiseze id-urile tuturor popasurilor vizitate de toti soiferii cu
93 | * salariul de 5000lei folosind implementarea DIVISION-ului cu dublu NOT
   |   ↳ EXISTS */
94 |
95 | SELECT DISTINCT id_popas
96 | FROM istoric_popasuri ip
97 | WHERE NOT EXISTS
98 |     (SELECT 1
99 |      FROM angajat a
100 |      WHERE tip_angajat = 'SOFER' and salariu = 5000

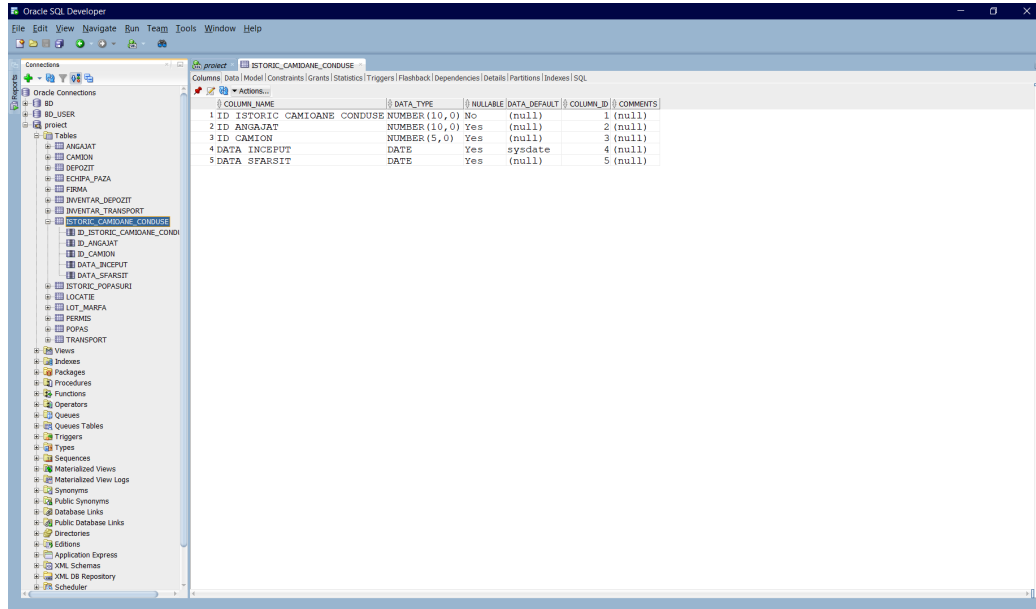
```

```

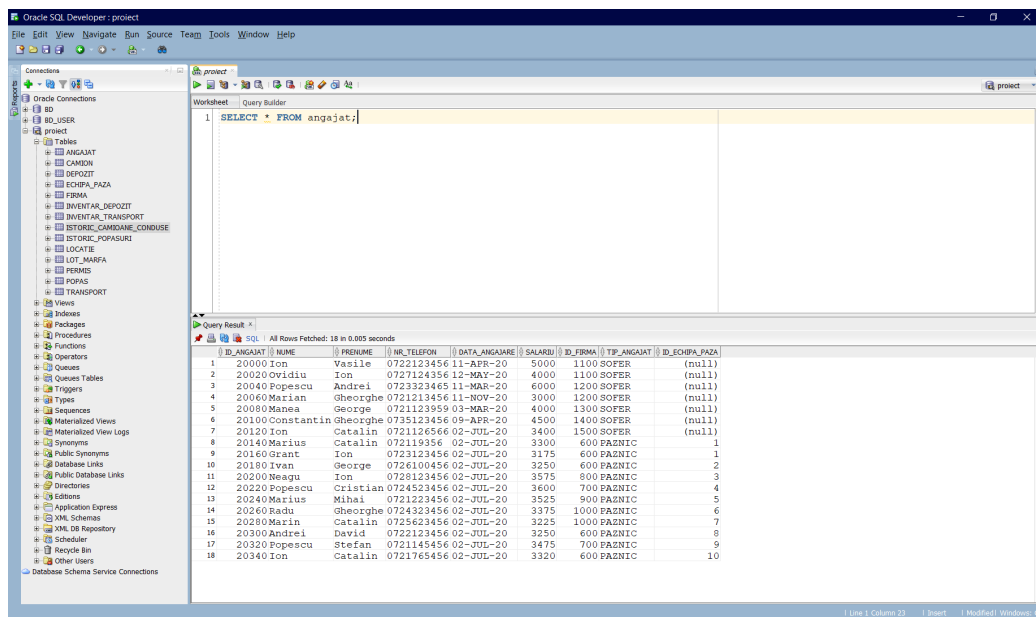
101 |         AND NOT EXISTS
102 |         (SELECT 'x'
103 |          FROM istoric_popasuri sip
104 |          WHERE a.id_angajat = sip.id_angajat
105 |                and sip.id_popas = ip.id_popas
106 |          )
107 |     );
108 |
109 | -- 3
110 |
111 | /* Sa se afiseze pentru totate popasurile vizitate de toti soferii cu
112 |    ↳ salariul
113 |    * de 5000 lei tipul popasului, numele sau si locatia unde se afla. Sa se
114 |    * utilizeze alta metoda de implementare fata de punctul anterior. */
115 | WITH sofer AS -- soferi cu salariu de 5000 lei
116 |     (SELECT id_angajat
117 |      FROM angajat
118 |      WHERE tip_angajat = 'SOFER' and salariu = 5000
119 |     ),
120 |
121 |     pop AS -- popasurile vizitate de toti soferii cu salariu de 5000 lei
122 |     (SELECT id_popas id_popas
123 |      FROM istoric_popasuri
124 |      WHERE id_angajat IN (SELECT *
125 |                          FROM sofer
126 |                          )
127 |      GROUP BY id_popas
128 |      HAVING count(id_angajat) = (SELECT count(*)
129 |                                FROM sofer
130 |                                )
131 |     )
132 |
133 | SELECT p.tip_popas "Tip popas", p.nume "Nume popas",
134 |        'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-') ||
135 |        ' Str. ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-')
136 |        ↳ "Locatie popas"
137 | FROM popas p JOIN pop ON (p.id_popas = pop.id_popas)
138 |              JOIN locatie l ON (p.id_locatie = l.id_locatie);

```

17 Dovezi rulare corecta a codului SQL



Tabele create



Date salvate in tabelul ANGAJAT

Oracle SQL Developer: project

Connections

- Oracle Connections
- project
 - BO_USER
 - Tables
 - ANGAJAT
 - CAMION
 - DEPOZIT
 - ECHIPA_FAZA
 - PRIMA
 - INVENTAR_DEPOZIT
 - INVENTAR_TRANSPORT
 - ISTORIC_CAMIONARE_CONDUSE
 - ISTORIC_POPASURI
 - LOCATIE
 - LOT_MARFA
 - PERMIS
 - POPAS
 - TRANSPORT
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues
 - Queues Tables
 - Triggers
 - Types
 - Sequences
 - Materialized Views
 - Materialized View Logs
 - Synonyms
 - Public Synonyms
 - Database Links
 - Public Database Links
 - Directories
 - Editions
 - Application Express
 - XML Schemas
 - XML DB Repository
 - Scheduler
 - Recycle Bin
 - Other Users
 - Database Schema Service Connections

Worksheet | Query Builder

1 SELECT * FROM transport;

Query Result: All Rows Fetched: 11 in 0.003 seconds

ID_TRANSPORT	DEPOZIT_PLECARE	DEPOZIT_DESTINATIE	ID_CAMION	DATA_PLECARE
1	10000	3000	500	15-06-MAY-21
2	10010	3000	500	25-08-MAY-21
3	10020	3500	1000	35-06-MAY-21
4	10030	4000	1000	45-06-MAY-21
5	10040	4500	1000	55-06-MAY-21
6	10050	5000	1500	115-07-MAY-21
7	10060	5000	1500	105-05-MAY-21
8	10070	3500	2000	85-06-MAY-21
9	10080	3500	2000	85-07-MAY-21
10	10090	3000	2500	25-10-MAY-21
11	10100	3000	2500	15-11-MAY-21

Line 1 Column 24 | Insert | Modified | Windows: C

Date salvate in tabelul TRANSPORT

Oracle SQL Developer: project

Connections

- Oracle Connections
- project
 - BO_USER
 - Tables
 - ANGAJAT
 - CAMION
 - DEPOZIT
 - ECHIPA_FAZA
 - PRIMA
 - INVENTAR_DEPOZIT
 - INVENTAR_TRANSPORT
 - ISTORIC_CAMIONARE_CONDUSE
 - ISTORIC_POPASURI
 - LOCATIE
 - LOT_MARFA
 - PERMIS
 - POPAS
 - TRANSPORT
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues
 - Queues Tables
 - Triggers
 - Types
 - Sequences
 - Materialized Views
 - Materialized View Logs
 - Synonyms
 - Public Synonyms
 - Database Links
 - Public Database Links
 - Directories
 - Editions
 - Application Express
 - XML Schemas
 - XML DB Repository
 - Scheduler
 - Recycle Bin
 - Other Users
 - Database Schema Service Connections

Worksheet | Query Builder

1 SELECT * FROM istoric_popasuri;

Query Result: All Rows Fetched: 10 in 0.002 seconds

ID_ISTORIC_POPAS	ID_POPAS	ID_ANGAJAT	ID_DATA_SOSIRE	DATA_PLECARE
1	50	180	20000	06-MAY-21 06-MAY-21
2	55	120	20000	06-MAY-21 06-MAY-21
3	60	180	20020	08-MAY-21 08-MAY-21
4	65	120	20020	08-MAY-21 08-MAY-21
5	70	100	20060	06-MAY-21 06-MAY-21
6	75	140	20060	06-MAY-21 06-MAY-21
7	80	140	20080	06-MAY-21 06-MAY-21
8	85	140	20100	06-MAY-21 06-MAY-21
9	90	180	20120	07-MAY-21 07-MAY-21
10	95	180	20120	05-MAY-21 05-MAY-21

Line 1 Column 31 | Insert | Modified | Windows: C

Date salvate in tabelul ISTORIC_POPASURI

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

27 WITH sps(cis) AS -- data primei opriri a unui sofer la Restaurantul Ceptura din Ploiesti
28 (SELECT to_char(min(sip.data_sosire), 'dd-mm-yyyy')
29 FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie)
30 JOIN istoric_popasuri sip ON (sp.id_popas = sip.id_popas)
31 WHERE initcap(sp.numa) = 'Restaurant Ceptura' and initcap(sl.localitate) = 'Ploiesti'
32 )
33
34 SELECT DECODE((SELECT upper(substr(sc.nr_inmatriculare, 0, 2))
35 FROM camion sc JOIN istoric_camioane_conduce sic ON (sc.id_camion = sic.id_camion)
36 JOIN angajat sa ON (sic.id_angajat = sa.id_angajat)
37 WHERE sic.id_angajat = a.id_angajat
38 and ip.data_sosire < NVL(sic.data_sfarsit, sysdate)
39 and ip.data_sosire > NVL(sic.data_inceput, sysdate)
40 ), 'PR', 'Din Prahova', 'Alta regiune'
41 ) "Verifica camion din Prahova",
42 a.numa "Nume angajat", a.prenume "Prenume angajat", a.salariu "Salariu",
43 p.numa "Nume popas", initcap(p.tip_popas) "Tip popas",
44
45 to_char(ip.data_sosire, 'hh24:mi') "Ora sosire",
46 substr(numtodsinterval((ip.data_plecure - ip.data_sosire), 'DAY'), 12, 5) "Durata",
47 (SELECT CASE WHEN salariu <= 3000 THEN salariu * 0.005
48 WHEN salariu <= 4500 THEN salariu * 0.01
49 )

```

Query Result 1 All Rows Fetched: 6 in 0.027 seconds

	Verifica camion din Prahova	Nume angajat	Prenume angajat	Salariu	Nume popas	Tip popas	Ora sosire	Durata	bani cheltuit	Locatie popas
1	Alta regiune	Constantin	Gheorghe	4500	Restaurant Grand	Restaurant	17:00	00:30	451ei	Jud: Iasi Loc: Iasi Str. Unirii Nr. 9991
2	Alta regiune	Ion	Vasile	5000	Restaurant Ceptura	Restaurant	21:00	00:30	751ei	Jud: Prahova Loc: Ploiesti Str. Lalelelor Nr. 142
3	Alta regiune	Ion	Vasile	5000	Conacul dintre vii	Motel	10:00	02:00	751ei	Jud: Constanta Loc: Constanta Str. Teilor Nr. 21
4	Alta regiune	Manea	George	4000	Restaurant Grand	Restaurant	17:00	00:30	401ei	Jud: Iasi Loc: Iasi Str. Unirii Nr. 9991
5	Din Prahova	Marian	Gheorghe	3000	Restaurant Grand	Restaurant	19:00	00:30	151ei	Jud: Iasi Loc: Iasi Str. Unirii Nr. 9991
6	Din Prahova	Marian	Gheorghe	3000	Motel Sunday	Motel	10:00	00:20	151ei	Jud: Prahova Loc: Ploiesti Str. Libertatii Nr. 13

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

Line 22 Column 7 Insert Modified Windows C

Ex. 11 - 1

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

79 WITH
80 avg_s AS -- media salarilor paznicilor per firma
81 (SELECT round(avg(salariu)) salariu, id_firma
82 FROM angajat
83 WHERE tip_angajat = 'PAZNIC'
84 GROUP BY id_firma
85 ),
86
87 cm_vizitat_popas AS -- id-ul celui mai vizitat popas
88 (
89 SELECT id_popas popas
90 FROM istoric_popasuri
91 GROUP BY id_popas
92 HAVING count(id_popas) = (SELECT max(count(id_popas))
93 FROM istoric_popasuri
94 GROUP BY id_popas
95 )
96 ),
97
98 loc_cm_vizitat_popas AS -- localitatea in care se afla cel mai vizitat popas
99 (SELECT sl.localitate localitate
100 FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie),
101 cm_vizitat_popas cmvp
102 WHERE sp.id_popas = cmvp.popas
103 )
104
105 SELECT a.numa "Nume paznic", a.prenume "Prenume paznic",

```

Query Result 1 All Rows Fetched: 3 in 0.029 seconds

	Nume paznic	Prenume paznic	Salariu dori	Cat de bine este platit	Vechime angajat	Nume firma	Locatie deplasament	Nr. Colegi
1	Marius	Mihai	3550	bine platit	329 zile 04:39	QUICK PROTECT	Jud: Constanta Loc: Constanta Str. Zorilor Nr. 13	Nu are colegi
2	Marius	Catalin	3400	bine platit	329 zile 04:39	STOP SEAL GUARD	Jud: Constanta Loc: Constanta Str. Victoriei Nr. 1231	coleg
3	Grant	Ion	3300	bine platit	329 zile 04:39	STOP SEAL GUARD	Jud: Constanta Loc: Constanta Str. Victoriei Nr. 1231	coleg

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

Line 84 Column 8 Insert Modified Windows C

Ex. 11 - 2

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet: Query Builder

```

158
159 WITH ac AS -- informatii despre cei mai bine platiti soferi din firma unde sunt angajati si camionul curent
160 (SELECT c.id_camion, c.marca, c.nr_inmatriculare, a.id_angajat, a.numa,
161      a.prenume, a.salariu, a.id_firma
162   FROM angajat a JOIN istoric_camioane conduse ic ON (a.id_angajat = ic.id_angajat)
163   JOIN camion c ON (c.id_camion = ic.id_camion)
164   WHERE (a.salariu, a.id_firma) in (SELECT max(salariu), id_firma -- obtine salariul mare din firma
165                                FROM angajat
166                                WHERE tip_angajat = 'SOFER'
167                                GROUP BY tip_angajat, id_firma
168                                )
169   and ic.data_sfarsit is null
170 )
171
172 SELECT
173   to_char(t.data_plecure, 'dd-mm-yyyy') "Data plecure",
174   to_char(t.data_plecure, 'hh24:mi') "Ora", -- data plecure transport
175   ac.numa "Nume sofer", ac.prenume "Prenume", -- cine a condus acest transport
176   ac.salariu "Salariu sofer", -- ce salariu are
177   DECODE((ac.salariu-3500)-abs(ac.salariu-3500)
178         , 0, 'bine platit', 'prost platit') "Cat de bine este platit",
179
180   ac.marca "Camion",
181   ac.nr_inmatriculare "Nr inmatriculare",

```

Query Result: All Rows Fetched: 7 in 0.057 seconds

Data plecure	Ora	Nume sofer	Prenume	Salariu sofer	Cat de bine este platit	Camion	Nr inmatriculare	Inmatriculat in judet	Proprietar depozit plecure	Locatie depozit plecure	Proprietar depozit destinatie	Locatie depozit destinatie
10-05-2021 06:00	06:00	Popescu	Andrei	6000	bine platit	MAN	IS01MAN	Iasi	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Deichmann	Loc: Constanta Str. Zorilor Nr. 13
20-05-2021 06:00	06:00	Constantin	Gheorghe	4500	bine platit	IVECO	MM13IVC	Maramures	Lexcom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
30-05-2021 06:00	06:00	Manea	George	4000	bine platit	DAF	B55ABD	Bucuresti	EasyCargo	Loc: Bucuresti Str. Munzii Nr. 1412	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
40-05-2021 08:00	08:00	Ion	Vasile	5000	bine platit	VOLVO	CT12FSD	Constanta	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Ploiesti Str. Primaverii Nr. 412
50-05-2021 06:00	06:00	Popescu	Andrei	6000	bine platit	MAN	IS01MAN	Iasi	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Deichmann	Loc: Constanta Str. Zorilor Nr. 13
60-05-2021 08:00	08:00	Ion	Catalin	3400	prost platit	SCANIA	B259CN	Bucuresti	FedEx	Loc: Bucuresti Str. I mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
71-05-2021 05:00	05:00	Ion	Vasile	5000	bine platit	VOLVO	CT12FSD	Constanta	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113

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

Line 182 Column 16 | Insert | Modified | Windows C

Ex. 11 - 3.1

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet: Query Builder

```

158
159 WITH ac AS -- informatii despre cei mai bine platiti soferi din firma unde sunt angajati si camionul curent
160 (SELECT c.id_camion, c.marca, c.nr_inmatriculare, a.id_angajat, a.numa,
161      a.prenume, a.salariu, a.id_firma
162   FROM angajat a JOIN istoric_camioane conduse ic ON (a.id_angajat = ic.id_angajat)
163   JOIN camion c ON (c.id_camion = ic.id_camion)
164   WHERE (a.salariu, a.id_firma) in (SELECT max(salariu), id_firma -- obtine salariul mare din firma
165                                FROM angajat
166                                WHERE tip_angajat = 'SOFER'
167                                GROUP BY tip_angajat, id_firma
168                                )
169   and ic.data_sfarsit is null
170 )
171
172 SELECT
173   to_char(t.data_plecure, 'dd-mm-yyyy') "Data plecure",
174   to_char(t.data_plecure, 'hh24:mi') "Ora", -- data plecure transport
175   ac.numa "Nume sofer", ac.prenume "Prenume", -- cine a condus acest transport
176   ac.salariu "Salariu sofer", -- ce salariu are
177   DECODE((ac.salariu-3500)-abs(ac.salariu-3500)
178         , 0, 'bine platit', 'prost platit') "Cat de bine este platit",
179
180   ac.marca "Camion",
181   ac.nr_inmatriculare "Nr inmatriculare",

```

Query Result: All Rows Fetched: 7 in 0.057 seconds

Salariu sofer	Cat de bine este platit	Camion	Nr inmatriculare	Inmatriculat in judet	Proprietar depozit plecure	Locatie depozit plecure	Proprietar depozit destinatie	Locatie depozit destinatie
1 6000	bine platit	MAN	IS01MAN	Iasi	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Deichmann	Loc: Constanta Str. Zorilor Nr. 13
2 4500	bine platit	IVECO	MM13IVC	Maramures	Lexcom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
3 4000	bine platit	DAF	B55ABD	Bucuresti	EasyCargo	Loc: Bucuresti Str. Munzii Nr. 1412	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
4 5000	bine platit	VOLVO	CT12FSD	Constanta	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Ploiesti Str. Primaverii Nr. 412
5 6000	bine platit	MAN	IS01MAN	Iasi	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Deichmann	Loc: Constanta Str. Zorilor Nr. 13
6 3400	prost platit	SCANIA	B259CN	Bucuresti	FedEx	Loc: Bucuresti Str. I mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
7 5000	bine platit	VOLVO	CT12FSD	Constanta	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113

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

Line 182 Column 16 | Insert | Modified | Windows C

Ex. 11 - 3.2

Oracle SQL Developer: project

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

224 WITH depozit_plecare AS
225 (SELECT id_depozit, sf.numr detinator, 'Loc: ' || NVL(sl.localitate, '-')
226      || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-') locatie
227 FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
228      JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
229 ),
230 depozit_destinatie AS
231 (SELECT id_depozit, sf.numr detinator, 'Loc: ' || NVL(sl.localitate, '-')
232      || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-') locatie
233 FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
234      JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
235 )
236
237 SELECT lm.numr "Marfa", lm.cantitate "Cantitate marfa",
238        DECODE(lower(lm.numr), 'carne miel', 'alimente', 'carne porc', 'alimente',
239              'carne pui', 'alimente', 'lapte', 'alimente', 'rosii', 'alimente',
240              'ardei', 'alimente', 'tricouri', 'imbracaminte', 'hanorace', 'imbracaminte',

```

Query Result

All Rows Fetched: 16 in 0.078 seconds

Marfa	Cantitate marfa	Tip aliment	Marca camion	Nr. inmatriculare	Inmatriculat in judetul	Data si ora plecarea	Nr. luni de la transport	Detinator depozit plecarea	Locatie depozit plecarea	Detinator depozit destinatie	Locatie depozit destinatie
1 tricouri	100	imbracaminte	VOLVO	MM57RXP	Maramures	05-05-2021 06:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
2 hanorace	50	imbracaminte	VOLVO	MM57RXP	Maramures	05-05-2021 06:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
3 rosii	1500	alimente	IVECO	MM131VC	Maramures	06-05-2021 06:00	1	Lextom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
4 lapte	120	alimente	IVECO	MM131VC	Maramures	06-05-2021 06:00	1	Lextom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
5 lapte	100	alimente	DAF	B55ASD	Bucuresti	06-05-2021 06:00	1	EasyCargo	Loc: Bucuresti Str. Muncii Nr. 1412	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
6 pantofi	20	incalzitaminte	MAN	IS01MAN	Iasi	06-05-2021 06:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Delchmann	Loc: Constanta Str. Zorilor Nr. 13
7 rosii	1000	alimente	SCANIA	PH138CN	Necunoscut	06-05-2021 08:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
8 ardei	500	alimente	SCANIA	PH138CN	Necunoscut	06-05-2021 08:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
9 carne porc	300	alimente	VOLVO	CT12FSD	Constanta	06-05-2021 08:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Tulcea Str. Victoriei Nr. 113
10 carne miel	100	alimente	VOLVO	CT12FSD	Constanta	06-05-2021 08:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Tulcea Str. Victoriei Nr. 113
11 pantofi	30	incalzitaminte	MAN	IS01MAN	Iasi	07-05-2021 06:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Delchmann	Loc: Constanta Str. Zorilor Nr. 13
12 blugi	35	imbracaminte	SCANIA	B298CN	Bucuresti	07-05-2021 08:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
13 carne pui	250	alimente	VOLVO	CT90MM	Constanta	08-05-2021 06:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113
14 rochii	50	imbracaminte	VOLVO	CT90MM	Constanta	10-05-2021 07:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113
15 fuste	35	imbracaminte	VOLVO	CT90MM	Constanta	10-05-2021 07:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113
16 hanorace	35	imbracaminte	VOLVO	CT12FSD	Constanta	11-05-2021 05:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113

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

Line 232 Column 29 | Insert | Modified | Windows | G

Ex. 11 - 4.1

Oracle SQL Developer: project

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

224 WITH depozit_plecare AS
225 (SELECT id_depozit, sf.numr detinator, 'Loc: ' || NVL(sl.localitate, '-')
226      || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-') locatie
227 FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
228      JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
229 ),
230 depozit_destinatie AS
231 (SELECT id_depozit, sf.numr detinator, 'Loc: ' || NVL(sl.localitate, '-')
232      || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-') locatie
233 FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
234      JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
235 )
236
237 SELECT lm.numr "Marfa", lm.cantitate "Cantitate marfa",
238        DECODE(lower(lm.numr), 'carne miel', 'alimente', 'carne porc', 'alimente',
239              'carne pui', 'alimente', 'lapte', 'alimente', 'rosii', 'alimente',
240              'ardei', 'alimente', 'tricouri', 'imbracaminte', 'hanorace', 'imbracaminte',

```

Query Result

All Rows Fetched: 16 in 0.078 seconds

Inmatriculare	Inmatriculat in judetul	Data si ora plecarea	Nr. luni de la transport	Detinator depozit plecarea	Locatie depozit plecarea	Detinator depozit destinatie	Locatie depozit destinatie
17RXP	Maramures	05-05-2021 06:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
27RXP	Maramures	05-05-2021 06:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
31VC	Maramures	06-05-2021 06:00	1	Lextom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
41VC	Maramures	06-05-2021 06:00	1	Lextom Trans Asd	Loc: Bucuresti Str. Unirii Nr. 212	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
5ASD	Bucuresti	06-05-2021 06:00	1	EasyCargo	Loc: Bucuresti Str. Muncii Nr. 1412	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
61MAN	Iasi	06-05-2021 06:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Delchmann	Loc: Constanta Str. Zorilor Nr. 13
73SCN	Necunoscut	06-05-2021 08:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
83SCN	Necunoscut	06-05-2021 08:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Profi	Loc: Iasi Str. Mihai Eminescu Nr. 991
92FSD	Constanta	06-05-2021 08:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Tulcea Str. Victoriei Nr. 113
102FSD	Constanta	06-05-2021 08:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Tulcea Str. Victoriei Nr. 113
111MAN	Iasi	07-05-2021 06:00	1	Transibo	Loc: Bucuresti Str. Libertatii Nr. 112	Delchmann	Loc: Constanta Str. Zorilor Nr. 13
123CN	Bucuresti	07-05-2021 08:00	1	FedEx	Loc: Bucuresti Str. 1 mai Nr. 1213	ABOUT YOU	Loc: Constanta Str. Victoriei Nr. 123
130MM	Constanta	08-05-2021 06:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	Lidl	Loc: Tulcea Str. Victoriei Nr. 113
140MM	Constanta	10-05-2021 07:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113
150MM	Constanta	10-05-2021 07:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113
162FSD	Constanta	11-05-2021 05:00	1	Road Logistics	Loc: Tulcea Str. Unirii Nr. 213	H and M	Loc: Tulcea Str. Victoriei Nr. 113

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

Line 232 Column 29 | Insert | Modified | Windows | G

Ex. 11 - 4.2

```

283 WITH soferi AS -- soferii care detin cel putin 2 categorii de permis auto si care
284 -- lucreaza la una din firmele: Road Logistics, Transibo, Lextom Trans Asd
285 (SELECT a.id_angajat id_sofer
286 FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
287 JOIN firma f ON (f.id_firma = a.id_firma)
288 WHERE initcap(f.ume) = 'Road Logistics' or initcap(f.ume) = 'Transibo'
289 or initcap(f.ume) = 'Lextom Trans Asd'
290 GROUP BY a.id_angajat
291 HAVING count(p.id_angajat) >= 2
292 )
293
294 SELECT a.ume "Nume sofer", a.prenume "Prenume",
295 CASE WHEN to_number(substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'), 2, 9)) <= 356 THEN 'INCEPATOR'
296 WHEN to_number(substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'), 2, 9)) <= 700 THEN 'AVANSAT'
297 ELSE 'PROFESIONIST'
298 END "Experienta",
299 DECODE((a.salariu-4000)-abs(a.salariu-4000), 0, 'bine platit', 'prost platit') "Cat de bine este platit",
300 c.marca "Marca camion", c.nr_inmatriculare "Nr. inmatriculare",
301 (SELECT sf.ume -- cine detine depozitul de plecare
302 FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
303 WHERE sd.id_depozit = t.depozit_plecare) "Proprietar depozit plecare",
304 (SELECT 'loc: ' || NVL(sl.localitate, '-') || ' Str. ' -- unde se afla depozitul
305 || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-')
306 FROM depozit sd JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
307 WHERE sd.id_depozit = t.depozit_plecare) "Locatie depozit plecare",
308
309 (SELECT sf.ume -- cine detine depozitul de destinatie

```

ID	Nume sofer	Prenume	Experienta	Cat de bine este platit	Marca camion	Nr. inmatriculare	Proprietar depozit plecare	Locatie depozit plecare	Proprietar depozit destinatie	Locatie depozit destinatie
1	Ion	Vasile	AVANSAT	bine platit	VOLVO	CT12FSD	Road Logistics	Loc: Tulcea Str. Unirii Nr.... Lidl	Loc: Floiesti Str. Primaverii Nr...	
2	Constantin	Gheorghe	AVANSAT	bine platit	IVECO	MM13IVC	Lextom Trans Asd	Loc: Bucuresti Str. Unirii ... Profi	Loc: Iasi Str. Mihai Eminescu Nr...	
3	Ion	Vasile	AVANSAT	bine platit	VOLVO	CT12FSD	Road Logistics	Loc: Tulcea Str. Unirii Nr.... H and M	Loc: Tulcea Str. Victoriei Nr. 113	

Ex. 11 - 5

```

345 (SELECT id_lot_marfa
346 FROM LOCATIE l JOIN DEPOZIT d ON (l.id_locatie = d.id_locatie)
347 JOIN INVENTAR_DEPOZIT id ON (d.id_depozit = id.id_depozit)
348 WHERE initcap(localitate) = 'Tulcea' AND initcap(strada)='Unirii' AND nr = 213
349 AND to_char(id.data_sosire, 'dd-mm-yyyy') = '02-04-2021'
350 );
351
352 /* stregem din baza de date popasurile care nu au fost vizitate pana acum de
353 * niciun sofer de tir */
354 DELETE FROM POPAS
355 WHERE id_popas not in (select id_popas FROM ISTORIC_POPASURI);
356
357 commit; -- salvam modificarile facute
358
359 /* 16. Formulati in limbaj natural si implementati in SQL: o cerere ce utilizeaza
360 * operatia outer-join pe minimum 4 tabele si doua cereri ce utilizeaza operatia
361 * division. */
362
363 -- outer join - versiunea 1 va fi optimizata la cerinta 17

```

5 rows updated.

1 row updated.

1 row deleted.

Commit complete.

Ex. 12 - 2 updateuri + 1 delete urmate de commit

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

570 WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin cel mult 2 permise auto
571 (SELECT p.id_angajat, a.numa, a.prenume, a.salariu
572 FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
573 WHERE a.tip_angajat = 'SOFER'
574 and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
575 GROUP BY p.id_angajat, a.numa, a.prenume, a.salariu
576 HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1 permis
577 )
578
579 SELECT c.marca, c.nr_inmatriculare, f.numa,
580 NVL(a.numa, '-') "Numa sofer", NVL(a.prenume, '-') "Prenume sofer",
581 NVL(to_char(a.salariu), '-') "Salariu sofer",
582 NVL(to_char(t.data_plecure, 'dd-mm-yyyy'), 'nu a realizat transporturi') "Data transport"
583 FROM camion c JOIN firma f ON (c.id_firma = f.id_firma)
584 FULL OUTER JOIN istoric_camioane_conduse icc ON (c.id_camion = icc.id_camion)
585 FULL OUTER JOIN ang a ON (icc.id_angajat = a.id_angajat)
586 FULL OUTER JOIN transport t ON (c.id_camion = t.id_camion)
587 WHERE
588 upper(c.marca) != 'IVECO' and
589 f.tip_firma = 'TRANSPORT' and
590 ((icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(icc.data_sfarsit, sysdate))
591 or t.data_plecure is null

```

Query Result: All Rows Fetched: 12 in 0.004 seconds

	MARCA	NR_INMATRICULARE	NUMA	Numa sofer	Prenume sofer	Salariu sofer	Data transport
1	VOLVO	CT90MM	Road Logistics Ovidiu Ion	4000	10-05-2021		
2	VOLVO	CT90MM	Road Logistics Ovidiu Ion	4000	08-05-2021		
3	SCANIA	PH13SCN	Transibo Marian Gheorghe	3000	06-05-2021		
4	DAF	B5SASD	EasyCargo Manea George	4000	06-05-2021		
5	SCANIA	B29SCN	FedEx Ion Catalin	3400	07-05-2021		
6	MAN	IS01MAN	Transibo Popescu Andrei	6000	07-05-2021		
7	MAN	IS01MAN	Transibo Popescu Andrei	6000	06-05-2021		
8	MAN	TN20QWE	Road Logistics Ion Vasile	5000	nu a realizat transporturi		
9	SCANIA	IS32ISA	Road Logistics Ion Vasile	5000	nu a realizat transporturi		
10	VOLVO	CT12FSD	Road Logistics Ion Vasile	5000	11-05-2021		
11	VOLVO	CT12FSD	Road Logistics Ion Vasile	5000	06-05-2021		
12	DAF	TM11IOP	EasyCargo	-	-	nu a realizat transporturi	

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

Line 386 Column 1 Insert Modified Windows C

Ex. 16 - 1v1 outer join

Oracle SQL Developer: proiect

File Edit View Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

516 WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin cel mult 2 permise auto
517 (SELECT p.id_angajat, a.numa, a.prenume, a.salariu
518 FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
519 WHERE a.tip_angajat = 'SOFER'
520 and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
521 GROUP BY p.id_angajat, a.numa, a.prenume, a.salariu
522 HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1 permis
523 )
524
525 SELECT icc.id_camion, icc.id_angajat, t.data_plecure, c.marca, c.nr_inmatriculare, c.id_firma
526 FROM camion c FULL OUTER JOIN istoric_camioane_conduse icc ON (c.id_camion = icc.id_camion)
527 FULL OUTER JOIN transport t ON (c.id_camion = t.id_camion)
528 WHERE c.marca != 'IVECO' and
529 (t.data_plecure is null or icc.data_inceput is null or
530 (icc.data_inceput < t.data_plecure and t.data_plecure <= NVL(icc.data_sfarsit, sysdate)))
531 )
532
533 SELECT tc.marca, tc.nr_inmatriculare, f.numa,
534 NVL(a.numa, '-') "Numa sofer", NVL(a.prenume, '-') "Prenume sofer",
535 NVL(to_char(a.salariu), '-') "Salariu sofer"

```

Query Result: All Rows Fetched: 12 in 0.017 seconds

	MARCA	NR_INMATRICULARE	NUMA	Numa sofer	Prenume sofer	Salariu sofer	Data transport
1	VOLVO	CT90MM	Road Logistics Ovidiu Ion	4000	10-05-2021		
2	VOLVO	CT90MM	Road Logistics Ovidiu Ion	4000	08-05-2021		
3	SCANIA	PH13SCN	Transibo Marian Gheorghe	3000	06-05-2021		
4	DAF	B5SASD	EasyCargo Manea George	4000	06-05-2021		
5	SCANIA	B29SCN	FedEx Ion Catalin	3400	07-05-2021		
6	MAN	IS01MAN	Transibo Popescu Andrei	6000	07-05-2021		
7	MAN	IS01MAN	Transibo Popescu Andrei	6000	06-05-2021		
8	MAN	TN20QWE	Road Logistics Ion Vasile	5000	nu a realizat transporturi		
9	SCANIA	IS32ISA	Road Logistics Ion Vasile	5000	nu a realizat transporturi		
10	VOLVO	CT12FSD	Road Logistics Ion Vasile	5000	11-05-2021		
11	VOLVO	CT12FSD	Road Logistics Ion Vasile	5000	06-05-2021		
12	DAF	TM11IOP	EasyCargo	-	-	nu a realizat transporturi	

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

Line 529 Column 19 Insert Modified Windows C

Ex. 17 - 1v2 outer join versiune optimizată

```

Oracle SQL Developer: project
File Edit View Navigate Run Source Team Tools Window Help

Worksheet | Query Builder
410 * dupa numeaprenume */
411
412 WITH marca_camion_ang AS -- marciile de camioane conduse de angajatul Ion Vasile
413 (select c.marca
414 from camion c join istoric_camioane_conduse icc on (c.id_camion = icc.id_camion)
415 join angajat a on (icc.id_angajat = a.id_angajat)
416 where initcap(a.num) = 'Ion' and initcap(a.prenume) = 'Vasile'
417 ),
418
419 division_ang AS -- id-urile angajatilor care conduc cel mult aceleasi marci de camioane precum ang Ion Vasile (inclusiv el)
420 (select icc.id_angajat id_angajat
421 from camion c join istoric_camioane_conduse icc on (c.id_camion = icc.id_camion)
422 where c.marca in
423 (SELECT *
424 FROM marca_camion_ang
425 )
426 group by icc.id_angajat
427 having count(*) <=
428 (SELECT count(*)
429 FROM marca_camion_ang
430 )
431
432 MINUS
433
434 select icc.id_angajat
435 from camion c join istoric_camioane_conduse icc on (c.id_camion = icc.id_camion)
436 where c.marca not in (SELECT *
437 FROM marca_camion_ang

```

Query Result 4

SQL | All Rows Fetched: 4 in 0.021 seconds

ID	NUME	PRENUME	SALARIU	Nr. permis auto
1	Popescu	Andrei	6000	1
2	Ovidiu	Ion	4000	1
3	Ion	Catalin	3400	1
4	Marian	Gheorghe	3000	1

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

Line 421 Column 19 | Insert | Modified | Windows: C

Ex. 16 - 2-1 division

```

Oracle SQL Developer: project
File Edit View Navigate Run Source Team Tools Window Help

Worksheet | Query Builder
443 JOIN permis p ON (a.id_angajat = p.id_angajat)
444 WHERE not (initcap(a.num) = 'Ion' and initcap(a.prenume) = 'Vasile')
445 GROUP BY a.num, a.prenume, a.salariu
446 ORDER BY 4 desc, 3 desc, 1, 2;
447
448 --- 2
449
450 /* Sa se afiseze id-urile tuturor popasurilor vizitate de toti soferii cu
451 * salariul de 5000lei folosind implementarea DIVISION-ului cu dublu NOT EXISTS */
452
453 SELECT DISTINCT id_popas
454 FROM istoric_popasuri ip
455 WHERE NOT EXISTS
456 (SELECT 1
457 FROM angajat a
458 WHERE tip_angajat = 'SOFER' and salariu = 5000
459 AND NOT EXISTS
460 (SELECT 'x'
461 FROM istoric_popasuri sip
462 WHERE a.id_angajat = sip.id_angajat
463 and sip.id_popas = ip.id_popas
464 )
465 );
466
467 --- 3
468
469 /* Sa se afiseze pentru toate popasurile vizitate de toti soferii cu salariul
470 * de 5000 lei tipul popasului, numele sau si locatia unde se afla. Sa se

```

Query Result 4

SQL | All Rows Fetched: 2 in 0.006 seconds

ID_POPAS
1
2

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

Line 462 Column 25 | Insert | Modified | Windows: C

Ex. 16 - 2-2 division

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query with line numbers 471 through 497. The query is a complex SQL statement involving subqueries and joins. The bottom panel shows the 'Query Result' with 2 rows fetched in 0.02 seconds. The result is a table with 3 columns: 'Tip popas', 'Nume popas', and 'Locatie popas'. The data rows are: 1. RESTAURANT, Restaurant Captura, Jud: Prahova, Loc: Floiesti Str. Lalelelor Nr. 142; 2. MOTEL, Conacul dintre vii, Jud: Constanta, Loc: Constanta Str. Teilor Nr. 21.

```

471 * utilizeze alta metoda de implementare fata de punctul anterior. */
472
473 WITH sofer AS -- soferi cu salariu de 5000 lei
474 (SELECT id_angajat
475 FROM angajat
476 WHERE tip_angajat = 'SOFER' and salariu = 5000
477 ),
478
479 pop AS -- popasurile vizitate de toti soferii cu salariu de 5000 lei
480 (SELECT id_popas id_popas
481 FROM istoric_popasuri
482 WHERE id_angajat IN (SELECT *
483 FROM sofer
484 )
485 GROUP BY id_popas
486 HAVING count(id_angajat) = (SELECT count(*)
487 FROM sofer
488 )
489 )
490
491 SELECT p.tip_popas "Tip popas", p.numa "Nume popas",
492 'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-') ||
493 ' Str: ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-') "Locatie popas"
494 FROM popas p JOIN pop ON (p.id_popas = pop.id_popas)
495 JOIN locatie l ON (p.id_locatie = l.id_locatie);
496
497

```

Tip popas	Nume popas	Locatie popas
1 RESTAURANT	Restaurant Captura	Jud: Prahova Loc: Floiesti Str. Lalelelor Nr. 142
2 MOTEL	Conacul dintre vii	Jud: Constanta Loc: Constanta Str. Teilor Nr. 21

Ex. 16 - 2-3 division