

# **PROIECT BAZE DE DATE**

## **ANUL I**

***Tema proiectului:*** Magazin de electronice

***Nume:*** Petre-Şoldan Adela

***Grupa:*** 131

***Seria:*** 13

## CUPRINS:

1) Descrierea modelului real și a regulilor de funcționare .....	3
2) Restricții .....	3
3) Entități .....	4
4) Descrierea relațiilor.....	6
5) Prezentarea atributelor.....	7
6) Diagrama entitate relație.....	10
7) Diagrama conceptuală.....	11
8) Schemele relaționale.....	12
9) Realizarea normalizării până la forma normală trei.....	12
10) Creare tabele.....	15
11) Inserare date.....	24
12) Interogări exercițiul 12.....	42
13) Operații de update și delete exercițiul 13.....	47
14) Interogări exercițiul 15.....	49
15) Denormalizarea si justificarea necesitatii acesteia.....	52

## **1) Descrierea modelului real și a regulilor de funcționare:**

Modelul de date urmărește activitatea unui lanț de magazine care se ocupă cu vânzarea electronicelor și care are mai multe sedii pe teritoriul țării. Un sediu are stabilit un contract pe durată nedeterminată cu mai mulți furnizori, aceștia ocupându-se de alimentarea sediului cu diverse produse. Un anumit produs nu poate fi preluat decât de un singur furnizor, astfel că se va găsi numai în sediile cu care furnizorul acestuia are contract.

Printre angajații firmei se numără administratorii și scenariștii. Un administrator este responsabil de gestionarea și bunăstarea unuia sau mai multor sedii, iar într-un sediu lucrează un singur administrator. Scenaristul este persoana care scrie textul pentru o reclamă, iar într-o reclamă este prezentat un singur produs. La o reclamă lucrează doar un scenarist. Fiecare scenarist are dreptul să își aleagă o zi liberă în cadrul săptămânii, detaliu prevăzut în contractul de angajare. Pentru fiecare reclamă reținem și costul final necesar realizării sale, pentru o imagine de ansamblu a pierderilor și câștigurilor magazinului.

Firma apelează la diverse metode de a se promova, printre care se numără organizarea de concursuri cu diverse premii. Pentru a se înscrie la un concurs, o persoană trebuie să își creeze cont pe site-ul firmei și să plătească o taxă de înscriere. Fiecare client are dreptul să își facă un singur cont, unde își va alege un nume de utilizator. Concursurile vor avea loc pe acest site, au întrebări grilă și nu durează mai mult de 20 de minute. Fiecare premiu are un punctaj minim necesar și constă într-un produs din cele furnizate, o sumă de bani sau amândouă. Concurentul va primi premiul cu cel mai mare punctaj minim mai mic sau egal cu punctajul obținut. Un concurs are mai multe premii, fiecare putând fi câștigat de mai multe persoane în funcție de scorurile obținute de acestea. Modelul oferă posibilitatea de a afla cine a participat și la ce concurs, precum și punctajul obținut. De aici se poate deduce și premiul câștigat de un concurent, sau dacă nu a câștigat nimic.

De asemenea, sunt disponibile și pachete promoționale care includ unul sau mai multe produse, nu neapărat de la același furnizor. Prețul unui astfel de pachet este mai redus decât suma prețurilor produselor componente și poate fi achiziționat dintr-un sediu numai dacă toate produsele componente sunt furnizate către sediul respectiv. În plus, la achiziționarea unui astfel de pachet, clientul primește niște puncte de fidelitate. Când acumulează cel puțin 50 de puncte, le poate converti în bani. Acești bani îi poate folosi doar în scopul achiziționării produselor din magazine. Un pachet promoțional nu poate aduce mai mult de 30 de puncte de fidelitate.

Modelul ține evidența vânzărilor de pachete promoționale, putând spune ce persoană a cumpărat un pachet și din ce sediu

## **2) Modelul funcționează pe baza mai multor restricții:**

- Un angajat nu poate fi în același timp și scenarist și administrator, dar un beneficiar poate să achiziționeze produse și să participe la concursuri.

- La o reclamă poate lucra un singur scenarist. Modelul nu își propune să urmărească activitatea celorlalți specialiști necesari pe care îi angajează firma pentru realizarea reclamei. Costul pentru o reclamă include și aceste cheltuieli.
- Fiecare sediu este alimentat de cel puțin un furnizor.
- În cazul angajaților s-a considerat un singur număr de telefon care poate fi atât fix cât și mobil, iar în cazul furnizorilor și clienților este reținută o singură adresă de e-mail. Adresa de e-mail a furnizorului va fi de fapt adresa unui reprezentant.
- Un pachet promoțional include minim un produs și maxim patru.
- Un premiu nu poate aparține mai multor concursuri.
- Toate contractele cu furnizorii au început înaintea monitorizării vânzărilor de pachete promoționale.
- Produsele au preț universal.
- Plata taxei de înscriere pentru un concurs se realizează online. Modelul nu ține evidența datelor acestor plăți, ci doar presupune că toți concurenții au plătit taxa respectivă.
- Firma a decis că un client nu poate achiziționa același pachet promoțional dintr-un sediu de mai multe ori, pentru a nu ieși în pierdere.
- Modelul nu urmărește sediul în care lucrează un scenarist, ci doar care este administratorul fiecărui sediu. De asemenea, se presupune că premiile câștigate sunt revendicate de clienți de la cel mai apropiat sediu disponibil sau livrate la domiciliul acestora.
- Dacă un client și-a creat cont pe platformă nu înseamnă că trebuie să participe la concursuri. Acest site are mai multe funcționalități în cadrul firmei care nu sunt urmărite de către diagrama, cum ar fi realizarea unei comenzi.
- Pentru a participa la concursuri, este necesar ca un beneficiar să își creeze cont pe platformă.
- Numele de utilizator trebuie să fie unic, asemenea CNP-ului, chiar dacă nu sunt chei primare.
- O reclamă prezintă un singur produs, iar un produs poate apărea în mai multe reclame. Există reclame care aduc în prim plan activitatea magazinului în general, astfel ca nu prezintă un produs specific. În acest caz cheia străină a produsului din tabela reclama va fi nulă.
- Suprafața unui sediu și costul final al unei reclame sunt estimate fără zecimale.
- Modelul ține doar evidența achizițiilor de pachete promoționale, nu și a produselor individuale, primele fiind mai importante datorită punctelor de fidelitate care trebuie calculate pentru fiecare cumpărător.

**3) Entitățile necesare modelului de date sunt:** PRODUS, CONCURS, PREMIU, FURNIZOR, SEDIU, UTILIZATOR, BENEFICIAR, PACHET\_PROMOTIONAL, RECLAMA și ANGAJAT, cu subentitățile SCENARIST și ADMINISTRATOR.

Toate entitățile sunt independente, cu excepția entității PREMIU care este dependentă de CONCURS.

În cele ce urmează vom descrie amănunțit semnificația fiecărei entități, precizând și care este cheia primară.

**FURNIZOR** = firma care se ocupă de fabricarea produselor și livrarea acestora în sediile cu care a stabilit un contract. Cheia primară se numește **cod\_furnizor**.

**SEDIU** = clădire care face parte din lanțul de magazine al firmei, care se ocupă cu vânzarea produselor aduse de furnizorii cu care are contract. Cheia primară se numește **cod\_sediu**.

**PRODUS** = bun material care provine de la un singur furnizor și transportat în toate sediile cu care furnizorul său a semnat un contract. Având în vedere tema proiectului, acest produs poate fi orice dispozitiv care funcționează pe bază de electricitate, ca de exemplu telefon mobil, laptop, aspirator sau uscător de păr. Cheia primară se numește **cod\_produs**.

**RECLAMA** = metoda de promovare ce constă într-un videoclip de scurtă durată care este publicat pe diverse canale de televiziune și care poate prezenta un produs care se poate găsi în sediile cu care furnizorul sau are contract. Cheia primară se numește **cod\_reclama**.

**CONCURS** = serie de întrebări grilă care se desfășoară pe site-ul firmei și durează maxim 20 de minute. Are o taxă de înscriere și oferă diverse premii. Cheia primară se numește **cod\_concurs**.

**PREMIU** = entitate dependentă de concurs, care constă într-o sumă de bani și/sau un produs dintre cele furnizate în cadrul firmei. Un premiu poate fi câștigat de mai multe persoane. Cheia primară este compusă din **cod\_concurs** și **cod\_premiu**.

**PACHET\_PROMOTIONAL** = ansamblu de maxim 5 produse care poate fi găsit în sediile în care se găsesc toate produsele componente. La cumpărarea sa, clientul primește un anumit număr de puncte de fidelitate. Cheia primară se numește **cod\_pachet**.

**ANGAJAT** = persoana angajată în cadrul firmei, care are diverse atribuții în funcție de titlul pe care îl ocupă. Cheia primară se numește **cod\_angajat**.

**SCENARIST** = subentitate a entității **ANGAJAT** care se ocupă de scrierea scenariului necesar pentru o reclamă. Cheia primară se numește **cod\_angajat**.

**ADMINISTRATOR** = cealaltă subentitate a entității **ANGAJAT**, care se ocupă de buna funcționare a sediului/sediilor în care lucrează și stabilește un anumit set de reguli pe care trebuie să îl urmeze angajații. Cheia primară se numește **cod\_angajat**.

**BENEFICIAR** = persoana care beneficiază de serviciile oferite de firmă, putând fi atât client cât și concurent simultan. Clientul este cel care cumpără un produs sau un pachet promoțional, iar concurent înseamnă participant la unul sau mai multe concursuri. Cheia primară se numește **cod\_beneficiar**.

**UTILIZATOR** = entitate care conține datele de autentificare ale beneficiarilor care și-au creat cont pe platformă. Cheia primară este **cod\_utilizator**.

#### 4) DESCRIEREA RELATIILOR:

Urmează o descriere a relațiilor prezente în modelul de date, cu precizarea cardinalității minime și maxime. Fiecare relație primește un nume sugestiv pe baza entităților legate:

**FURNIZOR\_furnizeaza\_PRODUS** = relație care leagă entitățile FURNIZOR și PRODUS, având rolul de a surprinde legătura dintre acestea (care este furnizorul unui produs). Aceasta are cardinalitatea minimă **1:1** (un furnizor trebuie să furnizeze cel puțin un produs și un produs trebuie furnizat de cel puțin un furnizor) și cardinalitatea maximă **1:n** (un furnizor poate furniza mai multe produse, dar un produs trebuie să fie furnizat de un singur furnizor).

**FURNIZOR\_alimenteaza\_sediu** = relație de tip many to many care leagă entitățile FURNIZOR și SEDIU (ce furnizori sunt responsabili de alimentarea căror sedii și au contract cu acestea). Relația are cardinalitatea minimă **1:0** (un furnizor nu trebuie să alimenteze niciun sediu, iar un sediu trebuie să fie alimentat de un furnizor) și cardinalitatea maximă **n:m** (un furnizor poate alimenta mai multe sedii, așa cum un sediu poate fi alimentat de mai mulți furnizori).

**ADMINISTRATOR\_administreaza\_SEDIU** = relație care leagă entitățile ADMINISTRATOR și SEDIU, având rolul de a surprinde legătura dintre acestea (cine este persoana care se ocupă de administrarea unui sediu). Relația are cardinalitatea minimă **1:0** (un administrator poate să nu administreze niciun sediu, dar un sediu trebuie să fie administrat de un singur administrator) și cardinalitatea maximă **n:1** (un administrator poate să administreze mai multe sedii, dar un sediu poate fi administrat de un singur administrator).

**SCENARIST\_creeaza\_RECLAMA** = relație care leagă entitățile SCENARIST și RECLAMA, având rolul de a surprinde legătura dintre acestea (cine este persoana care s-a ocupat de realizarea scenariului unei reclame). Relația are cardinalitatea minimă **1:0** (un scenarist poate să nu se ocupe de nicio reclamă, iar de o reclamă trebuie să se ocupe un singur scenarist) și cardinalitatea maximă **1:n** (un scenarist poate să creeze mai multe reclame, dar o reclamă poate fi creată de un singur scenarist).

**RECLAMA\_prezinta\_PRODUS** = relație care leagă entitățile RECLAMA și PRODUS (ce produs apare într-o reclamă). Relația are cardinalitatea minimă **0:0** și cardinalitatea maximă **n:1**.

**CONCURS\_ofera\_PREMIU** = relație care leagă entitățile CONCURS și PREMIU (ce premii sunt oferite și de care concurs). Deoarece fiecare concurs trebuie să aibă măcar un premiu, relația are cardinalitatea minimă **1:1** (un concurs trebuie să ofere cel puțin un premiu, iar un premiu trebuie să aparțină unui singur concurs) și cardinalitatea maximă **1:n** (un concurs poate să ofere mai multe premii, iar un premiu poate să aparțină unui singur concurs).

**PREMIU\_contine\_PRODUS** = relație care leagă entitățile PREMIU și PRODUS, având rolul de a surprinde legătura dintre acestea (ce produs conține fiecare premiu). Relația are cardinalitatea minimă **0:0** și cardinalitatea maximă **n:1**.

**UTILIZATOR\_participa\_la\_CONCURS** = relație de tip many to many care leagă entitățile UTILIZATOR și CONCURS (care utilizatori au participat la fiecare concurs). Relația are cardinalitatea minimă **0:0** (un utilizator poate să nu participe la niciun concurs, așa cum la un concurs poate să nu participe niciun utilizator) și cardinalitatea maximă **n:m** (la un concurs pot participa mai mulți utilizatori, iar un utilizator poate participa la mai multe concursuri)

**UTILIZATOR\_corespunde\_BENEFICIAR** = relație care leagă entitățile UTILIZATOR și BENEFICIAR, având rolul de a surprinde legătura dintre acestea (care este utilizatorul unui beneficiar în cazul în care are creat cont pe platformă). Relația are cardinalitatea minimă **0:1** și cardinalitatea maximă **1:1**.

**PACHET\_PROMOTIONAL\_contine\_PRODUS** = relație de tip many to many care leagă entitățile PACHET\_PROMOTIONAL și PRODUS (ce produse conține fiecare pachet promoțional). Relația are cardinalitatea minimă **0:1** (un pachet promoțional trebuie să includă cel puțin un produs, dar un produs nu trebuie să fie inclus într-un pachet promoțional) și cardinalitatea maximă **n:m** (un pachet promoțional poate include mai multe produse, așa cum un produs poate fi inclus în mai multe pachete promoționale).

**BENEFICIAR\_achizitioneaza\_PACHET\_PROMOTIONAL\_de\_la\_SEDIU** = relație de tip 3 care leagă entitățile BENEFICIAR, PACHET\_PROMOTIONAL și SEDIU, având denumirea "achizitioneaza". Relația permite modelului de date să stocheze informații despre ce persoană a cumpărat un pachet promoțional și din ce sediu.

## 5) PREZENTAREA ATRIBUTELOR:

**Entitatea FURNIZOR** prezintă următoarele atribute:

cod\_furnizor = variabilă de tip numeric (int), care reprezintă codul unui furnizor.

nume = variabilă de tip caracter, de lungime maximă 40 și care reprezintă numele furnizorului. Are valoare unică.

adresa\_mail\_repr = variabilă de tip caracter, de lungime maximă 40, care reprezintă adresa de e-mail a unui reprezentant al furnizorului.

**Entitatea PRODUS** are atributele:

cod\_produs = variabilă de tip numeric (int), care reprezintă codul unui furnizor.

cod\_furnizor = variabilă de tip numeric (int), care reprezintă codul furnizorului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul FURNIZOR și nu poate avea valoarea NULL.

nume = variabilă de tip caracter, de lungime maximă 40, semnificând numele produsului.

luni\_garantie = variabilă de tip numeric, de lungime maximă 2, reprezentând perioada de garanție a produsului exprimată în luni. Atributul este setat implicit cu valoarea 12, majoritatea produselor având această perioadă de garanție.

pret = variabilă de tip numeric, de lungime maximă 10, dintre care 2 cifre sunt zecimale. Reprezintă prețul produsului.

**Superentitatea ANGAJAT** are attributele:

cod\_angajat = variabilă de tip numeric (int), care reprezintă codul unui angajat.

nume = variabilă de tip caracter, de lungime maximă 25 și care reprezintă numele angajatului.

prenume = variabilă de tip caracter, de lungime maximă 25 și care reprezintă prenumele angajatului.

salariu = variabilă de tip numeric, de lungime maximă 7, reprezentând salariul angajatului exprimat în lei. Pentru că majoritatea angajaților au salariul de 3500 de lei, aceasta va fi valoarea implicită.

telefon = variabilă de tip caracter, de lungime maximă 15, reprezentând numărul de telefon al angajatului.

**Subentitatea SCENARIST** are attributele:

cod\_angajat = variabilă de tip numeric (int), reprezintă codului angajatului.

zi\_libera = variabilă de tip caracter, de lungime maximă 10, reprezentând ziua liberă pe care și-a ales-o angajatul (poate lua valorile luni, marți, miercuri, joi sau vineri).

experienta = variabilă de tip numeric, de lungime maximă 2 și care ia valori de la 1 la 10, reprezentând o aproximare a experienței câștigate de angajat până acum.

**Subentitatea ADMINISTRATOR** are attributele:

cod\_angajat = variabilă de tip numeric (int), reprezintă codul angajatului.

data\_angajare = variabilă de tip dată calendaristică, reprezentând data în care administratorul a fost pus în funcție în primul sediu din cadrul firmei.

**Entitatea RECLAMA** are attributele:

cod\_reclama = variabilă de tip numeric (int), care reprezintă codul unei reclame.

cod\_produs = variabilă de tip numeric (int), care reprezintă codul produsului prezentat de reclamă. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PRODUS și poate să aibă valoarea NULL.

cod\_angajat = variabilă de tip numeric (int), care reprezintă codul scenaristului care s-a ocupat de reclamă. Atributul trebuie să corespundă unei valori a cheii primare din tabelul SCENARIST și nu poate să aibă valoarea NULL.

durata = variabilă de tip numeric, de lungime maximă 3, reprezentând timpul de rulare al videoclipului reclamei exprimat în secunde.

cost\_total = variabilă de tip numeric, de lungime maximă 7, reprezentând costul necesar realizării reclamei. Acesta este exprimat în lei.

**Entitatea CONCURS** are attributele:

cod\_concurs = variabilă de tip numeric (int), care reprezintă codul unui concurs.

data\_concurs = variabilă de tip dată calendaristică, reprezentând data în care se desfășoară concursul.

durata = variabilă de tip numeric, de lungime maximă 2, reprezentând durata concursului exprimată în minute. Un concurs nu poate depăși 20 de minute.

subiect = variabilă de tip caracter, de lungime maximă 50, care reține despre ce vor fi întrebările din concurs.



taxa\_inscriere = variabilă de tip numeric, de lungime maximă 2, reprezentând taxa de înscriere exprimată în lei.

nr\_intrebări = variabilă de tip numeric, de lungime maximă 2, reprezentând numărul de întrebări.

**Entitatea dependentă PREMIU** are atributele:

cod\_premiu = variabilă de tip numeric (int), care reprezintă codul premiului

cod\_concurs = variabilă de tip numeric (int), care reprezintă codul concursului din care face parte premiul. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CONCURS și nu poate avea valoarea NULL.

punctaj\_minim = variabilă de tip numeric, de lungime maximă 3, care reprezintă punctajul necesar pentru a primi acest premiu. Ca restricție, nu poate fi mai mare de 100.

suma = variabilă de tip numeric, de lungime maximă 3, ce semnifică suma de bani inclusă în premiu

cod\_produs = variabilă de tip numeric (int), care reprezintă codul produsului inclus în premiu. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PRODUS și nu poate avea valoarea NULL.

**Entitatea PACHET\_PROMOTIONAL** are atributele:

cod\_pachet = variabilă de tip numeric (int), care reprezintă codul pachetului promoțional.

pret = variabilă de tip numeric, de lungime maximă 10 din care 2 cifre sunt zecimale. Reprezintă prețul pachetului promoțional.

puncte\_de\_fidelitate = variabilă de tip numeric, de lungime maximă 2, reprezentând punctele de fidelitate care pot fi câștigate la achiziționarea acestui pachet. Ca restricție, valoarea atributului nu poate depăși 30 de puncte.

**Entitatea SEDIU** are atributele: cod\_sediu, nume, cod\_administrator, adresa, suprafata (exprimată în metri patrati), adresa\_mail.

**Entitatea BENEFICIAR** are atributele: cod\_beneficiar, nume, preume, telefon și CNP. Asupra CNP-ului este aplicată o restricție de unicitate.

**Entitatea UTILIZATOR** are atributele: cod\_utilizator, cod\_beneficiar, nume\_utilizator, parola, data\_autentificare. Numele de utilizator trebuie să fie unic.

**Relația UTILIZATOR\_participa\_la\_CONCURS** are atributele:

cod\_concurs = variabilă de tip numeric (int), care reprezintă codul concursului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul CONCURS și nu poate să aibă valoarea NULL.

cod\_utilizator = variabilă de tip numeric (int), care reprezintă codul utilizatorului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul UTILIZATOR și nu poate să aibă valoarea NULL.

punctaj = variabilă de tip numeric, de lungime maximă 3, care reprezintă punctajul obținut de utilizator în cadrul concursului.

**Relația PRODUS\_contine\_PACHET\_PROMOTIONAL** are attributele:

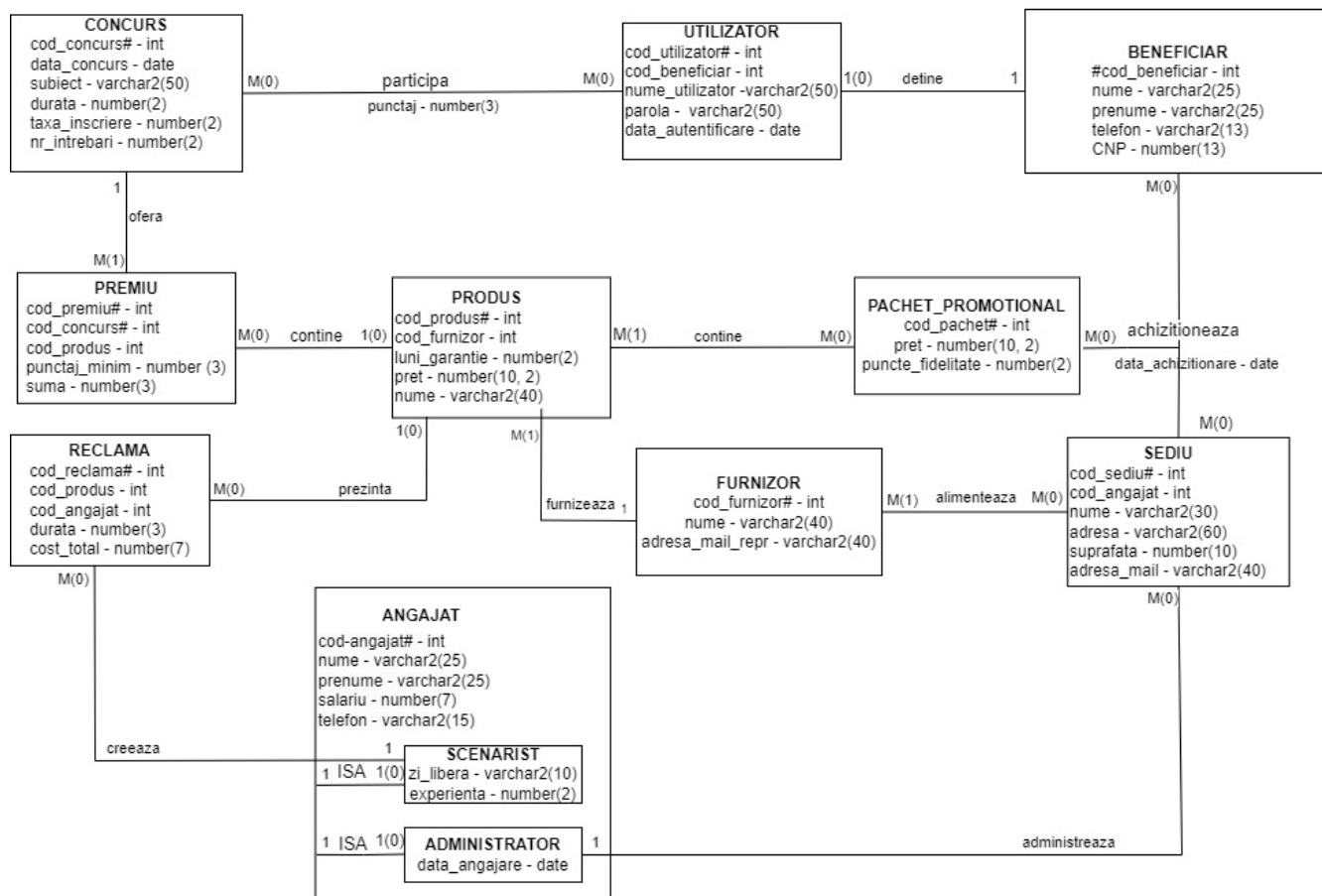
cod\_produs = variabilă de tip numeric (int), care reprezintă codul produsului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PRODUS și nu poate să aibă valoarea NULL.

cod\_pachet = variabilă de tip numeric (int), care reprezintă codul pachetului. Atributul trebuie să corespundă unei valori a cheii primare din tabelul PACHET\_PROMOTIONAL și nu poate să aibă valoarea NULL.

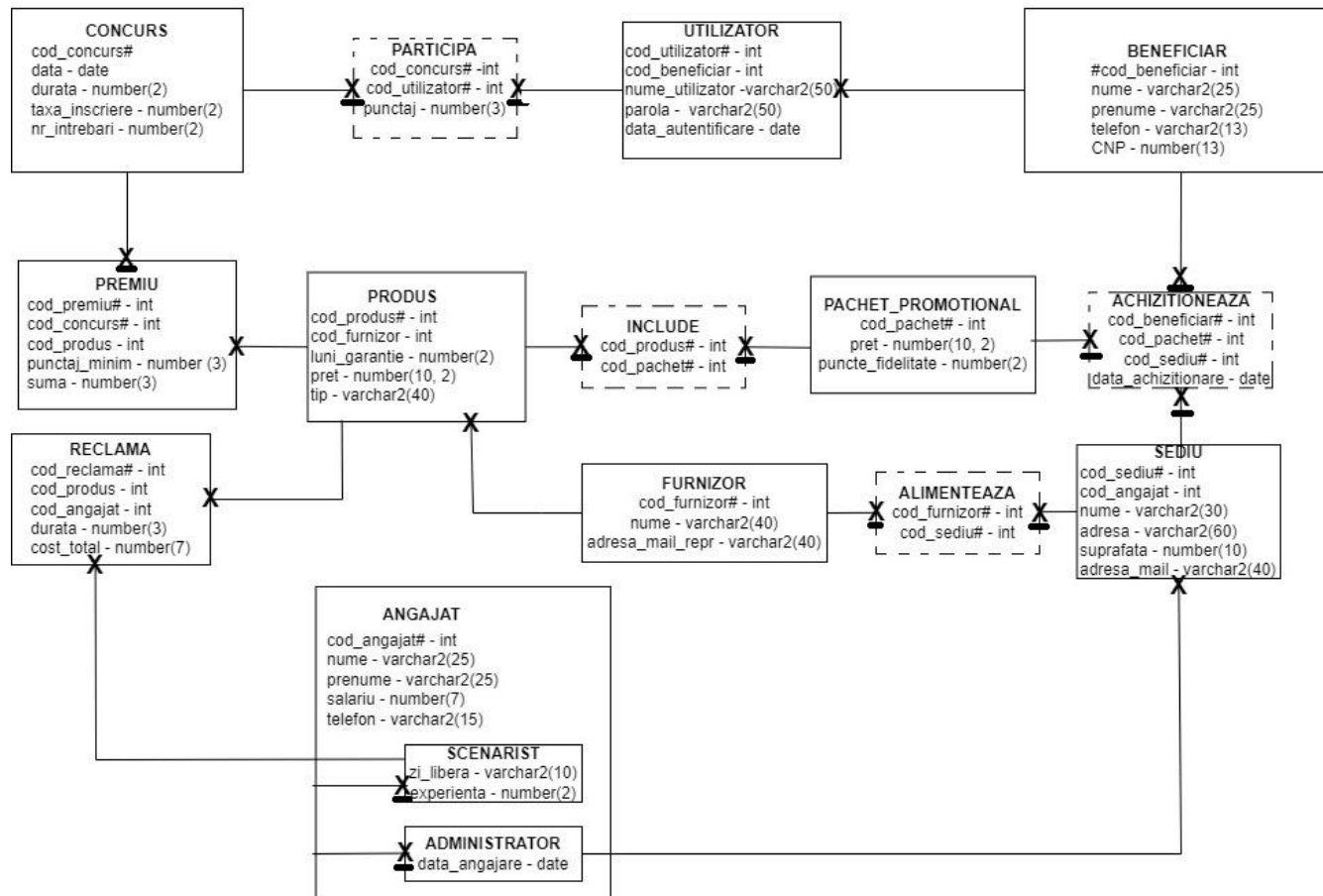
**Relația FURNIZOR\_alimenteaza\_SEDIU** are attributele: cod\_furnizor, cod\_sediu.

**Relația BENEFICIAR\_achizitioneaza\_PACHET\_PROMOTIONAL\_de\_la\_SEDIU** are attributele: cod\_beneficiar, cod\_pachet, cod\_sediu și data\_achizitionare. Ultimul atribut este de tip dată calendaristică și valoarea sa implicită este sysdate.

## 6) Diagrama Entitate Relație:



## 7)Diagrama Conceptuală:



## 8) Enumerarea schemelor relaționale:

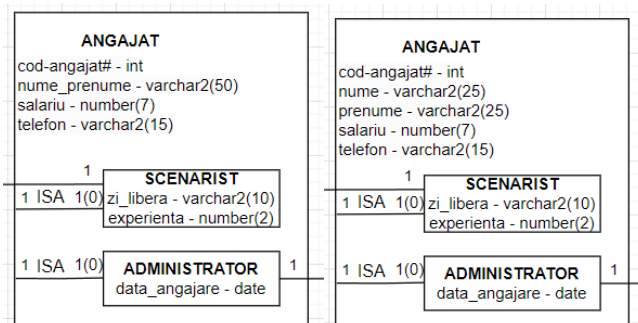
FURNIZOR(cod\_furnizor#, nume, adresa\_mail\_repr);  
PRODUS(cod\_produs#, cod\_furnizor, nume, luni\_garantie, pret);  
PACHET\_PROMOTIONAL(cod\_pachet#, pret, puncte\_fidelitate);  
ANGAJAT(cod\_angajat#, nume, prenume, salariu, telefon);  
SCENARIST(cod\_angajat#, zi\_libera, experienta);  
ADMINISTRATOR(cod\_angajat#, data\_angajare);  
SEDIU(cod\_sediu#, cod\_angajat, nume, adresa, suprafata, adresa\_mail);  
RECLAMA(cod\_reclama#, cod\_produs, cod\_angajat, durata, cost\_total);  
CONCURS(cod\_concurs#, data\_concurs, subiect, durata, nr\_intrebari, taxa\_inscriere);  
PREMIU(cod\_premiu#, cod\_concurs#, cod\_produs, punctaj\_minim, suma);  
BENEFICIAR(cod\_beneficiar#, nume, prenume, telefon, CNP);  
UTILIZATOR(cod\_utilizator#, cod\_beneficiar, nume\_utilizator, parola, data\_autentificare);  
PARTICIPA(cod\_concurs#, cod\_utilizator#, punctaj);  
CONTINE(cod\_produs#, cod\_pachet#);  
ALIMENTEAZA(cod\_furnizor#, cod\_sediu#);  
ACHIZITONEAZA(cod\_pachet#, cod\_beneficiar#, cod\_sediu#, data\_achizitionare);

## 9) Realizarea normalizării până la forma normală 3:

--- **Forma normală 1** presupune că toate atributele tabelelor să fie atomice, adică să stocheze o sigură informație. Spre exemplu, dacă am fi avut în tabela ANGAJAT atributul „nume\_prenume” în loc de atributele nume și prenume, acesta nu ar mai fi fost atomic și ar fi fost nevoie de operații în plus pentru a determina numele și prenumele. Relatia ANGAJAT(cod\_angajat#, nume\_prenume, salariu, telefon) nu este în prima forma normală din cauza atributului nume\_prenume care nu este atomic. După normalizare, tabela devine ANGAJAT(cod\_angajat#, nume, prenume, salariu, telefon).

Spre exemplu, în loc să avem atributul nume\_prenume cu valoarea „Popescu Irina”, vom avea atributele nume și prenume cu valorile „Popescu” și „Irina”.

Acest lucru este valabil și pentru tabela beneficiar, unde se regăsesc de asemenea atributele nume și prenume. În stânga avem o tabelă care nu este în fn1, iar în dreapta se respectă regula.

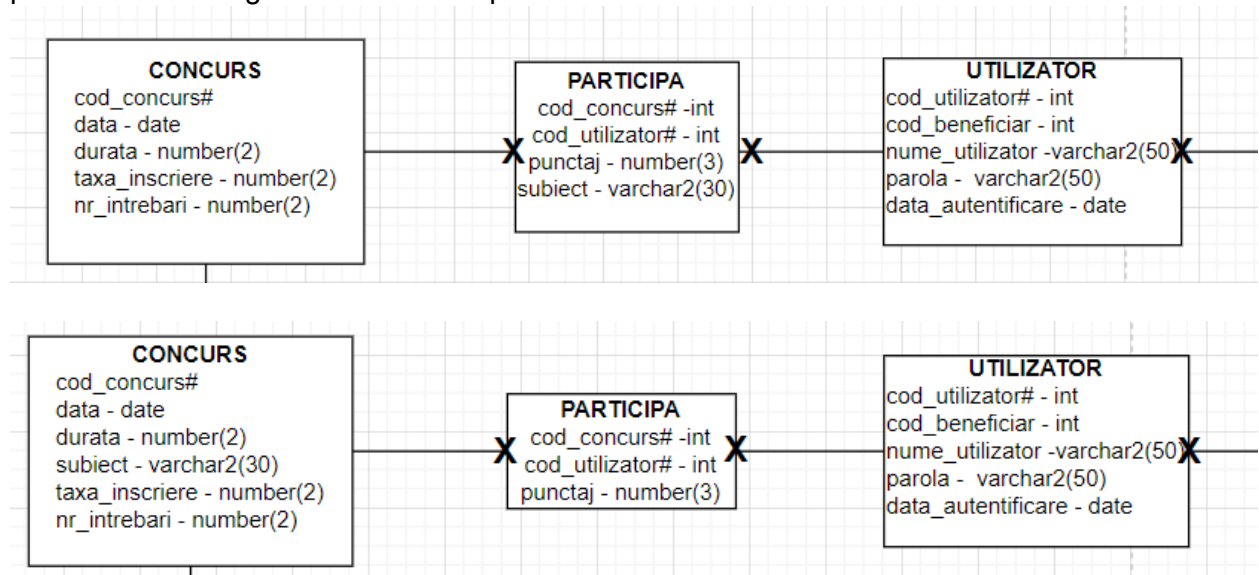


---**Forma normală 2** presupune, pe lângă forma normală 1, un atribut care nu este cheie dintr-o tabelă să depindă de întreaga cheie primară, nu doar de unele părți din ea.

Spre exemplu, în tabela asociativă "PARTICIPA" reținem codurile participantului și concursului (cele două formând cheia primară compusă), cât și punctajul obținut. Acest punctaj depinde în egală măsură și de concurs și de utilizatorul participant, neputând fi stocată în tabela UTILIZATOR sau CONCURS. Totuși, dacă pe lângă cheia primară compusă și atributul punctaj am fi avut, spre exemplu, atributul "subiect" plasat în tabela CONCURS, nu s-ar fi respectat forma normală 2 pentru că subiectul nu are nicio legătură cu entitatea UTILIZATOR, doar cu entitatea CONCURS.

Astfel, relația UTILIZATOR\_participa\_la\_CONCURS(cod\_concurs#, cod\_utilizator#, punctaj, subiect) este în forma normală 1, având toate atributele atomice, dar nu se află și în FN2. Pentru a realiza trecerea în FN2, relația devine UTILIZATOR\_participa\_la\_CONCURS(cod\_concurs#, cod\_utilizator#, punctaj), iar atributul subiect este mutat în tabela concurs.

La fel se întâmplă și pentru tabela ACHIZITIONEAZA, unde data achiziționării depinde de toate cele 3 (SEDIU, BENEFICIAR, PACHET\_PROMOTIONAL). Nu putem să adăugăm în tabela asociativă informații dintr-unul dintre cele 3 tabele în afară de cheia primară. Doar în figura a doua se respectă forma normală 2.



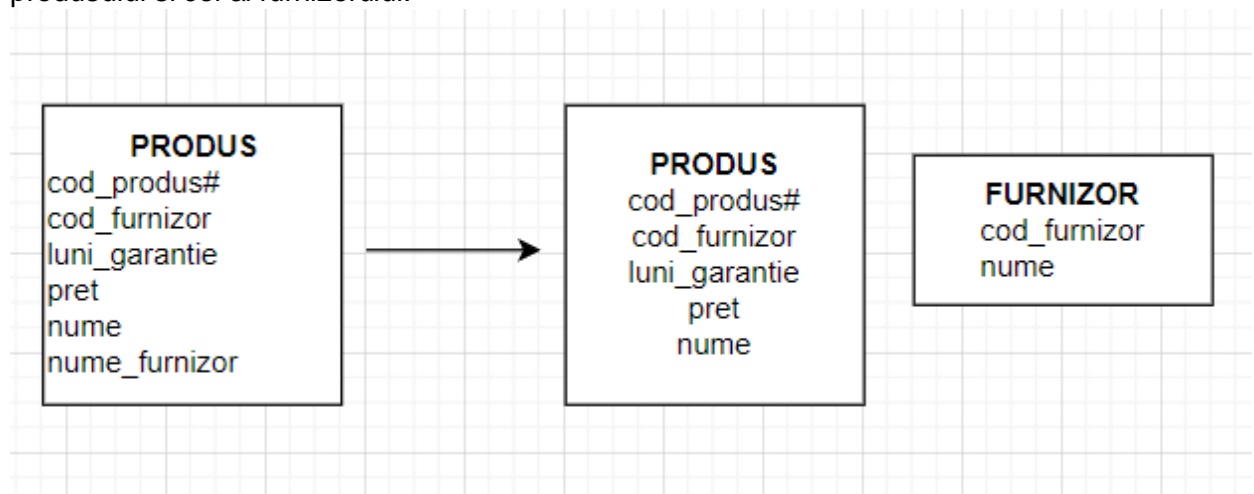
Presupunând ca nu ar fi existat deja tabela CONCURS, iar toate atributele sale erau plasate în relația PARTICIPA(cod\_concurs#, cod\_utilizator#, punctaj, subiect, data, durata, taxa\_inscriere, nr\_intrebari), aceasta din urmă s-ar fi descompus prin regula Casey-Delobel în 2 proiectii, PARTICIPA și CONCURS. Multimea de atribute dependente doar de cod\_concurs se va regăsi numai în tabela CONCURS.

--**Forma normală 3** presupune respectarea primelor 2 forme normale si, in plus, fiecare atribut care nu este cheie nu e dependent tranzitiv de nicio cheie. Altfel spus, fiecare atribut care nu face parte din cheie trebuie sa depinda de intreaga cheie si numai de aceasta.

Presupunem ca avem relatia PRODUS(cod\_produs#, cod\_furnizor, luni\_garantie, pret, nume, nume\_furnizor). Aceasta se afla in FN2 pentru ca orice atribut non-cheie este dependent de intreaga cheie primara cod\_produs. Totusi, relatia nu este si in FN3 deoarece numele furnizorului depinde tranzitiv de cheia primara prin intermediul atributului cod\_furnizor.

Pentru a o transforma in FN3 trebuie sa aplicam regula Casey-Delobel.

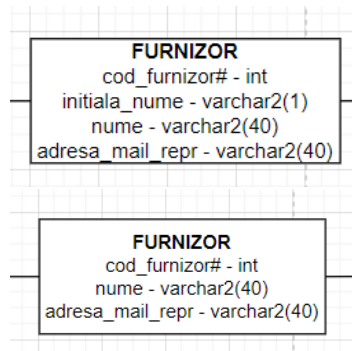
Fie relatia R (K, X1, X2, X3), avand cheia primara K si atributul X3 dependent tranzitiv de K prin X2:  $K \rightarrow X2 \rightarrow X3$  si  $X2 \rightarrow X3$ . R nu este in FN3, deci trebuie inlocuit cu doua proiectii R' (K, X1, X2) si R'' (X2, X3). Inlocuirea se face fara pierderi de informatie. Astfel, descompunem relatia in proiectiile PRODUS(cod\_produs#, cod\_furnizor, luni\_garantie, pret, nume) si FURNIZOR(cod\_furnizor, nume). Astfel, se elimina dependentele functionale tranzitive. Atributul nume\_furnizor poate fi trecut doar ca „nume” in tabela FURNIZOR, deoarece nu se mai poate crea confuzia dintre numele produsului si cel al furnizorului.



Relatia PRODUS este acum in FN3 deoarece toate attributele non-cheie depind de intreaga cheie primara si numai de aceasta. Acest lucru este valabil si pentru tabela FURNIZOR, unde numele furnizorului depinde numai de codul furnizorului, adica de cheia primara. In tabela FURNIZOR se pot adauga mai multe attribute dependente numai de cod\_furnizor (exemplu: adresa de e-mail a unui reprezentant) care altfel s-ar fi aflat in tabela PRODUS si ar fi creat si mai multe dependente tranzitive prin cod\_furnizor.

De asemenea, NF3 interzice situatia in care în tabela FURNIZOR am fi avut și un atribut care să rețină inițiala numelui furnizorului, s-ar fi creat redundanță, deoarece inițiala poate fi aflate din atributul “nume”. Un exemplu mai puțin evident ar fi stocarea simultană în tabela BENEFICIAR a CNP-ului și a datei de naștere, cea de-a doua

putându-se afla din primul atribut. Exemplul al doilea respectă regulile celei de a treia forme normale.



## 10) CREAREA TABELELOR:

```
create table FURNIZOR (  
cod_furnizor int constraint cod_furnizor_pk primary key,  
nume varchar2(40) constraint nume_furnizor_nn NOT NULL,  
adresa_mail_repr varchar2(40),  
constraint nume_furnizor_uq unique(nume)  
);
```

```
create table CONCURS (  
cod_concurs int constraint cod_concurs_pk primary key,  
data_concurs date default sysdate constraint data_concurs_nn NOT NULL,  
subiect varchar2(50) constraint subiect_nn NOT NULL,  
durata number(2) constraint durata_concurs NOT NULL,  
nr_intrebari number(2) default 20 constraint nr_intrebari_nn NOT NULL,  
taxa_inscriere number(2) constraint taxa_inscriere_nn NOT NULL,  
constraint durata_concurs_ck check(durata <= 20),  
constraint subiect_uq unique(subiect)  
);
```

```
create table BENEFICIAR (  
cod_beneficiar int,  
nume varchar2(25) constraint nume_beneficiar_nn NOT NULL,  
prenume varchar2(25) constraint prenume_beneficiar_nn NOT NULL,  
telefon varchar2(13),  
CNP number(13) constraint cnp_beneficiar_nn NOT NULL constraint cnp_beneficiar_uq  
unique,  
constraint cod_beneficiar_pk primary key(cod_beneficiar)  
);
```

```

create table PACHET_PROMOTIONAL (
cod_pachet int,
pret number(10, 2) constraint pret_pachet_nn NOT NULL,
puncte_fidelitate number(2),
constraint puncte_fidelitate_ck check(puncte_fidelitate <= 30),
constraint cod_pachet_pk primary key(cod_pachet)
);

```

```

create table PRODUS (
cod_produs int,
cod_furnizor int constraint produs_furnizor_nn NOT NULL,
nume varchar2(40) constraint nume_produs_nn NOT NULL,
luni_garantie number(2) default 12,
pret number(10, 2) constraint pret_produs_nn NOT NULL,
constraint cod_produs_pk primary key(cod_produs),
constraint nume_produs_uq unique(nume),
constraint include_furnizor_fk foreign key(cod_furnizor) references
FURNIZOR(cod_furnizor)
);

```

```

create table CONTINE (
cod_produs int,
cod_pachet int,
constraint contine_pk primary key(cod_produs, cod_pachet),
constraint contine_produs_fk foreign key(cod_produs) references
PRODUS(cod_produs),
constraint contine_pachet_fk foreign key(cod_pachet) references
PACHET_PROMOTIONAL(cod_pachet)
);

```

```

create table UTILIZATOR (
cod_utilizator int,
cod_beneficiar int constraint cod_beneficiar_nn NOT NULL,
nume_utilizator varchar2(50) constraint nume_utilizator_nn NOT NULL,
parola varchar2(50) constraint parola_nn NOT NULL,
data_autenticare date default sysdate constraint data_autenticare_nn NOT NULL,
constraint nume_utilizator_uq unique(nume_utilizator),
constraint cod_utilizator_pk primary key(cod_utilizator),
constraint cod_beneficiar_uq unique(cod_beneficiar),
constraint cod_beneficiar_fk foreign key(cod_beneficiar) references
BENEFICIAR(cod_beneficiar)
);

```



```

create table PARTICIPA (
cod_concurs int,
cod_utilizator int,
punctaj number(3) constraint punctaj_nn NOT NULL,
constraint participa_pk primary key(cod_concurs, cod_utilizator),
constraint participa_concurs_fk foreign key(cod_concurs) references
CONCURS(cod_concurs),
constraint participa_utilizator_fk foreign key(cod_utilizator) references
UTILIZATOR(cod_utilizator)
);

```

```

create table PREMIU (
cod_premiu int,
cod_concurs int constraint premiu_concurs_nn NOT NULL,
cod_produs int,
punctaj_minim number(3) constraint punctaj_minim_nn NOT NULL,
suma number(3) default 0,
constraint premiu_pk primary key(cod_premiu, cod_concurs),
constraint premiu_concurs_fk foreign key(cod_concurs) references
CONCURS(cod_concurs),
constraint premiu_produs_fk foreign key(cod_produs) references PRODUS(cod_produs)
);

```

```

create table ANGAJAT (
    cod_angajat int,
    nume varchar2(25) constraint nume_angajat_nn NOT NULL,
    prenume varchar2(25) constraint prenume_angajat_nn NOT NULL,
    salariu number(7) default 3500,
    telefon varchar2(15),
    constraint cod_angajat_pk primary key(cod_angajat)
);

```

```

create table SCENARIST (
    cod_angajat int constraint cod_scenarist_pk primary key references
    ANGAJAT(cod_angajat),
    zi_libera varchar2(10),
    experienta number(2)
);

```

```

create table ADMINISTRATOR (
    cod_angajat int constraint cod_administrator_pk primary key references
    ANGAJAT(cod_angajat),
    data_angajare date default sysdate constraint data_angajare_nn NOT NULL
);

```

```

create table RECLAMA (
cod_reclama int constraint cod_reclama_pk primary key,

```

```

cod_produs int,
cod_angajat int constraint cod_angajat_nn NOT NULL,
durata number(3) constraint durata_reclama_nn NOT NULL,
cost_total number(7) constraint cost_total_nn NOT NULL,
constraint reclama_produs_fk foreign key(cod_produs) references
PRODUS(cod_produs),
constraint reclama_scenarist_fk foreign key(cod_angajat) references
SCENARIST(cod_angajat)
);

create table SEDIU (
cod_sediu int constraint cod_sediu_pk primary key,
cod_angajat int constraint sediu_angajat_nn NOT NULL,
nume varchar2(30) constraint nume_sediu_nn NOT NULL,
adresa varchar2(60) constraint adresa_sediu_nn NOT NULL,
suprafata number(10) default 100000 constraint suprafata_sediu_nn NOT NULL,
adresa_mail varchar2(40),
constraint nume_sediu_uq unique(nume),
constraint sediu_administrator_fk foreign key(cod_angajat) references
ADMINISTRATOR(cod_angajat)
);

create table ACHIZITIONEAZA (
cod_beneficiar int,
cod_pachet int,
cod_sediu int,
data_achizitionare date default sysdate,
constraint achizitioneaza_pk primary key(cod_beneficiar, cod_pachet, cod_sediu),
constraint achizitioneaza_pachet__fk foreign key(cod_pachet) references
PACHET_PROMOTIONAL(cod_pachet),
constraint achizitioneaza_beneficiar__fk foreign key(cod_beneficiar) references
BENEFICIAR(cod_beneficiar),
constraint achizitioneaza_sediu__fk foreign key(cod_sediu) references
SEDIU(cod_sediu)
);

create table ALIMENTEAZA (
cod_furnizor int,
cod_sediu int,
constraint alimenteaza_pk primary key(cod_furnizor, cod_sediu),
constraint alimenteaza_sediu__fk foreign key(cod_sediu) references SEDIU(cod_sediu),
constraint alimenteaza_furnizor_fk foreign key(cod_furnizor) references
FURNIZOR(cod_furnizor)
);

```

```
--Create table.
create table FURNIZOR (
cod_furnizor int constraint cod_furnizor_pk primary key,
nume varchar2(40) constraint nume_furnizor_nn NOT NULL,
adresa_mail_repr varchar2(40),
constraint nume_furnizor_uq unique(nume)
);

create table CONCURS (
cod_concurs int constraint cod_concurs_pk primary key,
data_concurs date default sysdate constraint data_concurs_nn NOT NULL,
subiect varchar2(50) constraint subiect_nn NOT NULL,
durata number(2) constraint durata_concurs NOT NULL,
nr_intrebari number(2) default 20 constraint nr_intrebari_nn NOT NULL,
taxa_inscriere number(2) constraint taxa_inscriere_nn NOT NULL,
constraint durata_concurs_ck check(durata <= 20),
constraint subiect_uq unique(subiect)
);

create table BENEFICIAR (
cod_beneficiar int,
nume varchar2(25) constraint nume_beneficiar_nn NOT NULL,
prenume varchar2(25) constraint prenume_beneficiar_nn NOT NULL,
telefon varchar2(13),
CNP number(13) constraint cnp_beneficiar_nn NOT NULL constraint cnp_beneficiar_uq unique,
constraint cod_beneficiar_pk primary key(cod_beneficiar)
);

create table PACHET_PROMOTIONAL (
cod_pachet int,
pret number(10, 2) constraint pret_pachet_nn NOT NULL,
puncte_fidelitate number(2),
constraint puncte_fidelitate_ck check(puncte_fidelitate <= 30),
constraint cod_pachet_pk primary key(cod_pachet)
);
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x .. x x x

Task completed in 0.357 seconds

Table FURNIZOR created.

Table CONCURS created.

Table BENEFICIAR created.

```
create table PACHET_PROMOTIONAL (
  cod_pachet int,
  pret number(10, 2) constraint pret_pachet_nn NOT NULL,
  puncte_fidelitate number(2),
  constraint puncte_fidelitate_ck check(puncte_fidelitate <= 30),
  constraint cod_pachet_pk primary key(cod_pachet)
);

create table PRODUS (
  cod_produs int,
  cod_furnizor int constraint produs_furnizor_nn NOT NULL,
  nume varchar2(40) constraint nume_produs_nn NOT NULL,
  luni_garantie number(2) default 12,
  pret number(10, 2) constraint pret_produs_nn NOT NULL,
  constraint cod_produs_pk primary key(cod_produs),
  constraint nume_produs_uq unique(nume),
  constraint include_furnizor_fk foreign key(cod_furnizor) references FURNIZOR(cod_furnizor)
);

create table CONTINE (
  cod_produs int,
  cod_pachet int,
  constraint contine_pk primary key(cod_produs, cod_pachet),
  constraint contine_produs_fk foreign key(cod_produs) references PRODUS(cod_produs),
  constraint contine_pachet_fk foreign key(cod_pachet) references PACHET_PROMOTIONAL(cod_pachet)
);
```

Script Output x

Task completed in 0.179 seconds

Table FURNIZOR created.

Table CONCURS created.

Table BENEFICIAR created.

Table PACHET\_PROMOTIONAL created.

Table PRODUS created.

Table CONTINE created.

```

);
create table UTILIZATOR (
cod_utilizator int,
cod_beneficiar int constraint cod_beneficiar_nn NOT NULL,
nume_utilizator varchar2(50) constraint nume_utilizator_nn NOT NULL,
parola varchar2(50) constraint parola_nn NOT NULL,
data_autenticare date default sysdate constraint data_autenticare_nn NOT NULL,
constraint nume_utilizator_uq unique(nume_utilizator),
constraint cod_utilizator_pk primary key(cod_utilizator),
constraint cod_beneficiar_uq unique(cod_beneficiar),
constraint cod_beneficiar_fk foreign key(cod_beneficiar) references BENEFICIAR(cod_beneficiar)
);
create table PARTICIPA (
cod_concurs int,
cod_utilizator int,
punctaj number(3) constraint punctaj_nn NOT NULL,
constraint participa_pk primary key(cod_concurs, cod_utilizator),
constraint participa_concurs_fk foreign key(cod_concurs) references CONCURS(cod_concurs),
constraint participa_utilizator_fk foreign key(cod_utilizator) references UTILIZATOR(cod_utilizator)
);
create table PREMIU (
cod_premiu int,
cod_concurs int constraint premiu_concurs_nn NOT NULL,
cod_produs int,
punctaj_minim number(3) constraint punctaj_minim_nn NOT NULL,
suma number(3) default 0,
constraint premiu_pk primary key(cod_premiu, cod_concurs),
constraint premiu_concurs_fk foreign key(cod_concurs) references CONCURS(cod_concurs),
constraint premiu_produs_fk foreign key(cod_produs) references PRODUS(cod_produs)
);

```

Script Output x

Task completed in 0.218 seconds

Table CONTINE created.

Table UTILIZATOR created.

Table PARTICIPA created.

Table PREMIU created.

```

create table ANGAJAT (
    cod_angajat int,
    nume varchar2(25) constraint nume_angajat_nn NOT NULL,
    prenume varchar2(25) constraint prenume_angajat_nn NOT NULL,
    salariu number(7) default 3500,
    telefon varchar2(15),
    constraint cod_angajat_pk primary key(cod_angajat)
);

create table SCENARIST (
    cod_angajat int constraint cod_scenarist_pk primary key references ANGAJAT(cod_angajat),
    zi_libera varchar2(10),
    experienta number(2)
);

create table ADMINISTRATOR (
    cod_angajat int constraint cod_administrator_pk primary key references ANGAJAT(cod_angajat),
    data_angajare date default sysdate constraint data_angajare_nn NOT NULL
);

create table RECLAMA (
    cod_reclama int constraint cod_reclama_pk primary key,
    cod_produus int,
    cod_angajat int constraint cod_angajat_nn NOT NULL,
    durata number(3) constraint durata_reclama_nn NOT NULL,
    cost_total number(7) constraint cost_total_nn NOT NULL,
    constraint reclama_produus_fk foreign key(cod_produus) references PRODUS(cod_produus),
    constraint reclama_scenarist_fk foreign key(cod_angajat) references SCENARIST(cod_angajat)
);

```

Script Output x

Task completed in 0.137 seconds

Table ANGAJAT created.

Table SCENARIST created.

Table ADMINISTRATOR created.

Table RECLAMA created.

```

create table SEDIU (
  cod_sediu int constraint cod_sediu_pk primary key,
  cod_angajat int constraint sediu_angajat_nn NOT NULL,
  nume varchar2(30) constraint nume_sediu_nn NOT NULL,
  adresa varchar2(60) constraint adresa_sediu_nn NOT NULL,
  suprafata number(10) default 100000 constraint suprafata_sediu_nn NOT NULL,
  adresa_mail varchar2(40),
  constraint nume_sediu_uq unique(nume),
  constraint sediu_administrator_fk foreign key(cod_angajat) references ADMINISTRATOR(cod_angajat)
);

create table ACHIZITIONEAZA (
  cod_beneficiar int,
  cod_pachet int,
  cod_sediu int,
  data_achizitionare date default sysdate,
  constraint achizitioneaza_pk primary key(cod_beneficiar, cod_pachet, cod_sediu),
  constraint achizitioneaza_pachet_fk foreign key(cod_pachet) references PACHET_PROMOTIONAL(cod_pachet),
  constraint achizitioneaza_beneficiar_fk foreign key(cod_beneficiar) references BENEFICIAR(cod_beneficiar),
  constraint achizitioneaza_sediu_fk foreign key(cod_sediu) references SEDIU(cod_sediu)
);

create table ALIMENTEAZA (
  cod_furnizor int,
  cod_sediu int,
  constraint alimenteaza_pk primary key(cod_furnizor, cod_sediu),
  constraint alimenteaza_sediu_fk foreign key(cod_sediu) references SEDIU(cod_sediu),
  constraint alimenteaza_furnizor_fk foreign key(cod_furnizor) references FURNIZOR(cod_furnizor)
);

```

Script Output x

Task completed in 0.097 seconds

Table SEDIU created.

Table ACHIZITIONEAZA created.

Table ALIMENTEAZA created.

## 11) INSERARE DATE:

```
create sequence seq_furnizor
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;
```

```
insert into furnizor (cod_furnizor, nume, adresa_mail_repr)values(seq_furnizor.nextval,
'Electronice high quality', 'popescu.ioan@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr)values(seq_furnizor.nextval,
'Electric party', 'ionescu.maria@yahoo.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr)values(seq_furnizor.nextval,
'Electronics++', 'marinescu.mirela@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr)values(seq_furnizor.nextval,
'IT Direct', 'paul.enescu45@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr)values(seq_furnizor.nextval,
'Ultra circuite', NULL);
select * from furnizor;
```

```
create sequence seq_concurs
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;
```

```
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari,
taxa_inscriere)values(seq_concurs.nextval, to_date('12-04-2023', 'dd-mm-yyyy'),
'Biologie', 10, 15, 20);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari,
taxa_inscriere)values(seq_concurs.nextval, to_date('18-04-2023', 'dd-mm-yyyy'),
'Matematica', 20, 20, 30);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari,
taxa_inscriere)values(seq_concurs.nextval, to_date('27-02-2023', 'dd-mm-yyyy'),
'Literatura', 10, 16, 10);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari,
taxa_inscriere)values(seq_concurs.nextval, to_date('14-11-2022', 'dd-mm-yyyy'),
'Diverse', 7, 8, 10);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari,
taxa_inscriere)values(seq_concurs.nextval, to_date('15-01-2023', 'dd-mm-yyyy'),
'Vulcani', 13, 15, 25);
select * from concurs;
```



```
create sequence seq_beneficiar
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;
```

```
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Andronic', 'Marcel', '+400733456753',
5001210208170);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Suditu', 'Mara', '+400745839485', 2990310209242);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Antonescu', 'David', '+400748201839',
5010410205807);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Lupu', 'Eugen', '+400745678542', 1990915204669);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Alexandru', 'Miruna', '+400706543678',
6001130203816);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon,
CNP)values(seq_beneficiar.nextval, 'Munteanu', 'Ana', '+400733456784',
6010413100131);
select * from beneficiar;
```

```
create sequence seq_produs
start with 100
increment by 1
maxvalue 1000
nocycle
nocache;
```

```
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 100, 'Casti Wireless', 12, 50);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 100, 'Bec ultra', 24, 20);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 100, 'Incarcator telefon', 16, 40);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 110, 'Lanterna', 36, 50);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 110, 'Incarcator tableta', NULL, 10);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 120, 'Aspirator robot', 10, 300);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 120, 'Fier de calcat', 12, 120);
```

```

insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 130, 'Uscator de par', 12, 90);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 130, 'Uscator de par Pro', 12, 190);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 140, 'Telefon', 14, 3000);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 140, 'Laptop', 14, 6000);
insert into produs (cod_produs, cod_furnizor, nume, luni_garantie,
pret)values(seq_produs.nextval, 140, 'Tableta', 15, 1000);
select * from produs;

```

```

create sequence seq_pachet
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

```

```

insert into pachet_promotional (cod_pachet, pret,
puncte_fidelitate)values(seq_pachet.nextval, 70, 10);
insert into pachet_promotional (cod_pachet, pret,
puncte_fidelitate)values(seq_pachet.nextval, 80, 12);
insert into pachet_promotional (cod_pachet, pret,
puncte_fidelitate)values(seq_pachet.nextval, 6000, 30);
insert into pachet_promotional (cod_pachet, pret,
puncte_fidelitate)values(seq_pachet.nextval, 420, 20);
insert into pachet_promotional (cod_pachet, pret,
puncte_fidelitate)values(seq_pachet.nextval, 50, 3);
select* from pachet_promotional;

```

```

insert into contine (cod_produs, cod_pachet) values(100, 100);
insert into contine (cod_produs, cod_pachet) values(103, 100);
insert into contine (cod_produs, cod_pachet) values(104, 100);
insert into contine (cod_produs, cod_pachet) values(101, 110);
insert into contine (cod_produs, cod_pachet) values(107, 110);
insert into contine (cod_produs, cod_pachet) values(109, 120);
insert into contine (cod_produs, cod_pachet) values(110, 120);
insert into contine (cod_produs, cod_pachet) values(105, 120);
insert into contine (cod_produs, cod_pachet) values(105, 130);
insert into contine (cod_produs, cod_pachet) values(107, 130);
insert into contine (cod_produs, cod_pachet) values(103, 130);
insert into contine (cod_produs, cod_pachet) values(106, 130);
insert into contine (cod_produs, cod_pachet) values(111, 140);
insert into contine (cod_produs, cod_pachet) values(104, 140);
select * from contine;

```

```

create sequence seq_utilizator
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

```

```

insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola,
data_autenticare)values(seq_utilizator.nextval, 100, 'marcel233', 'f123ds', to_date('12-
05-2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola,
data_autenticare)values(seq_utilizator.nextval, 110, 'marra4', '35323d', to_date('25-04-
2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola,
data_autenticare)values(seq_utilizator.nextval, 120, 'dvd34_a', 'parola', to_date('30-05-
2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola,
data_autenticare)values(seq_utilizator.nextval, 130, 'mira__12', 'prdfgd', to_date('12-04-
2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola,
data_autenticare)values(seq_utilizator.nextval, 150, 'mnt_ana', '567fg', to_date('02-05-
2022', 'dd-mm-yyyy'));
select* from utilizator;

```

```

create sequence seq_premiu
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

```

```

insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 100, 103, 60, 20);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 100, 100, 70, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 100, 108, 90, 40);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 110, 106, 80, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 110, 103, 90, 35);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 120, NULL, 70, 210);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 120, 102, 75, 20);

```

```

insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 130, 101, 80, NULL);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 130, 105, 99, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 140, 111, 100, 0);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim,
suma)values(seq_premiu.nextval, 140, 104, 50, 10);
select* from premiu;

```

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 110, 50);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 100, 66);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 130, 74);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 140, 95);

```

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 100, 92);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 110, 94);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 130, 88);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 140, 50);

```

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 100, 97);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 110, 73);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 140, 60);

```

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 100, 99);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 110, 90);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 130, 79);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 140, 73);

```

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 100, 99);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 110, 45);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 130, 74);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 140, 100);
select* from participa;

```

```

create sequence seq_angajat
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

```

```

insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Lopataru', 'Alexandra', 3500, '+400758465748');

```

```

insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Cujbescu', 'Marius', 3600, '+400745678954');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Serban', 'Alina', 3700, '+400767584957');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Mihai', 'Costin', 3650, '+400778965435');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Lazar', 'Simon', 3700, '+400789657453');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Moldovan', 'Andreea', 5000, '+400709876547');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Oprea', 'Pavel', 5500, '+400798765467');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Dumitrescu', 'Bianca', 5600, '+400785768594');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Dinu', 'Dobrin', 5400, '+400734567895');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Cristea', 'Cristina', 5400, NULL);
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Todorescu', 'Simon', 6000, '+400734567845');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values
(seq_angajat.nextval, 'Anontescu', 'Briana', 7000, '+400787657893');
select* from angajat;

```

```

insert into scenarist (cod_angajat, zi_libera, experienta) values (100, 'miercuri', 6);
insert into scenarist (cod_angajat, zi_libera, experienta) values (110, 'luni', 8);
insert into scenarist (cod_angajat, zi_libera, experienta) values (120, 'luni', 10);
insert into scenarist (cod_angajat, zi_libera, experienta) values (130, 'joi', 7);
insert into scenarist (cod_angajat, zi_libera, experienta) values (140, 'vineri', 6);
select * from scenarist;

```

```

insert into administrator (cod_angajat, data_angajare) values (150, to_date('15-08-2021',
'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (160, to_date('12-09-2021',
'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (170, to_date('19-08-2022',
'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (180, to_date('02-11-2022',
'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (190, to_date('16-05-2023',
'dd-mm-yyyy'));
select* from administrator;

```

```
create sequence seq_reclama
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;
```

```
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 100, 140, 30, 2150);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 100, 130, 40, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, NULL, 130, 40, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 101, 110, 50, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 101, 130, 60, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 101, 140, 10, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 102, 110, 20, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 102, 140, 30, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 102, 130, 35, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 103, 120, 35, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 103, 110, 35, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 105, 120, 30, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 105, 140, 45, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 106, 130, 24, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 106, 120, 45, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 107, 110, 30, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 109, 140, 30, 500);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 110, 130, 45, 4000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 111, 110, 55, 5000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 110, 120, 30, 4030);
```

```

insert into reclama (cod_reclama, cod_produș, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, NULL, 120, 42, 2000);
insert into reclama (cod_reclama, cod_produș, cod_angajat, durata, cost_total) values
(seq_reclama.nextval, 108, 130, 25, 600);
select * from reclama;

```

```

create sequence seq_sediu
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

```

```

insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values
(seq_sediu.nextval, 150, 'Electric center', 'Bucuresti Sud', 10000,
'electric.center@gmail.com');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values
(seq_sediu.nextval, 160, 'Cable power', 'Centru Bucuresti City', 12000,
'cable.power234@yahoo.ro');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values
(seq_sediu.nextval, 170, 'Atacul tehnologiei', 'Ploiesti Periferie', 5000,
'technology.attack578@yahoo.com');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values
(seq_sediu.nextval, 170, 'IT Party', 'Buzau Cartier Dorobanti', 6000,
'partyit45@gmail.ro');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values
(seq_sediu.nextval, 190, 'ELECTRO LIGHT MAIN', 'Bucuresti Strada Pacii', 12000,
'electrowizBucharest@gmail.com');
select* from sediu;

```

```

insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 130);

```

```

insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 100);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 140);

```

```

insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 140);

```

```

insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 130);
insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 140);

```

```
insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 100);
insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 140);
select* from alimenteaza;
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (100, 100, 110, to_date('16-09-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (100, 100, 120, to_date('17-08-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (100, 130, 120, to_date('15-10-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (100, 140, 100, to_date('14-11-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (100, 120, 140, to_date('12-12-2022', 'dd-mm-yyyy'));
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 100, 120, to_date('13-08-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 110, 120, to_date('16-07-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 110, 130, to_date('17-05-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 120, 140, to_date('18-03-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 130, 120, to_date('19-02-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (110, 140, 110, to_date('20-01-2020', 'dd-mm-yyyy'));
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (120, 110, 120, to_date('21-11-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (120, 120, 140, to_date('22-12-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (120, 140, 100, to_date('23-04-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (120, 130, 120, to_date('24-04-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (120, 100, 110, to_date('25-06-2021', 'dd-mm-yyyy'));
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (130, 100, 120, to_date('26-06-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (130, 120, 140, to_date('27-07-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (130, 110, 130, to_date('28-09-2021', 'dd-mm-yyyy'));
```



```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (130, 140, 110, to_date('29-08-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (130, 110, 120, to_date('30-10-2019', 'dd-mm-yyyy'));
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (140, 110, 120, to_date('25-03-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (140, 130, 120, to_date('01-03-2019', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (140, 120, 140, to_date('02-09-2021', 'dd-mm-yyyy'));
```

```
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 100, 110, to_date('02-06-2018', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 110, 120, to_date('03-06-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 120, 110, to_date('04-07-2018', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 120, 140, to_date('05-07-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 130, 120, to_date('06-08-2019', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare)
values (150, 140, 110, to_date('07-08-2021', 'dd-mm-yyyy'));
select * from achizitioneaza;
```

```
create sequence seq_furnizor
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into furnizor (cod_furnizor, nume, adresa_mail_repr) values(seq_furnizor.nextval, 'Electronice high quality', 'popescu.ioan@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr) values(seq_furnizor.nextval, 'Electrtic party', 'ionescu.maria@yahoo.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr) values(seq_furnizor.nextval, 'Electronics++', 'marinescu.mirela@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr) values(seq_furnizor.nextval, 'IT Direct', 'paul.enescu45@gmail.com');
insert into furnizor (cod_furnizor, nume, adresa_mail_repr) values(seq_furnizor.nextval, 'Ultra circuite', NULL);
select * from furnizor;
```

```
create sequence seq_concurs
start with 100
increment by 10
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x

SQL | All Rows Fetched: 5 in 0.013 seconds

COD_FURNIZOR	NUME	ADRESA_MAIL_REPR
100	Electronice high quality	popescu.ioan@gmail.com
110	Electrtic party	ionescu.maria@yahoo.com
120	Electronics++	marinescu.mirela@gmail.com
130	IT Direct	paul.enescu45@gmail.com
140	Ultra circuite	(null)

```

create sequence seq_concurs
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari, taxa_inscriere)values(seq_concurs.nextval, to_date('12-04-2023', 'dd-mm-yyyy'), 'Biologie', 10, 15, 20);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari, taxa_inscriere)values(seq_concurs.nextval, to_date('18-04-2023', 'dd-mm-yyyy'), 'Matematica', 20, 20, 30);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari, taxa_inscriere)values(seq_concurs.nextval, to_date('27-02-2023', 'dd-mm-yyyy'), 'Literatura', 10, 16, 10);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari, taxa_inscriere)values(seq_concurs.nextval, to_date('14-11-2022', 'dd-mm-yyyy'), 'Diverse', 7, 8, 10);
insert into concurs (cod_concurs, data_concurs, subiect, durata, nr_intrebari, taxa_inscriere)values(seq_concurs.nextval, to_date('15-01-2023', 'dd-mm-yyyy'), 'Vulcani', 13, 15, 25);
select * from concurs;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result 6 x

SQL | All Rows Fetched: 5 in 0.019 seconds

	COD_CONCURS	DATA_CONCURS	SUBIECT	DURATA	NR_INTREBARI	TAXA_INSCRIERE
1	100	12-APR-23	Biologie	10	15	20
2	110	18-APR-23	Matematica	20	20	30
3	120	27-FEB-23	Literatura	10	16	10
4	130	14-NOV-22	Diverse	7	8	10
5	140	15-JAN-23	Vulcani	13	15	25

```

create sequence seq_beneficiar
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Andronic', 'Marcel', '+400733456753', 5001210208170);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Suditu', 'Mara', '+400745839485', 2990310209242);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Antonescu', 'David', '+400748201839', 5010410205807);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Lupu', 'Eugen', '+400745678542', 1990915204669);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Alexandru', 'Miruna', '+400706543678', 6001130203816);
insert into beneficiar (cod_beneficiar, nume, prenume, telefon, CNP)values(seq_beneficiar.nextval, 'Munteanu', 'Ana', '+400733456784', 6010413100131);
select * from beneficiar;

```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0.048 seconds

	COD_BENEFICIAR	NUME	PRENUME	TELEFON	CNP
1	100	Andronic	Marcel	+400733456753	5001210208170
2	110	Suditu	Mara	+400745839485	2990310209242
3	120	Antonescu	David	+400748201839	5010410205807
4	130	Lupu	Eugen	+400745678542	1990915204669
5	140	Alexandru	Miruna	+400706543678	6001130203816
6	150	Munteanu	Ana	+400733456784	6010413100131

```

create sequence seq_produș
start with 100
increment by 1
maxvalue 1000
nocycle
nocache;

insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 100, 'Casti Wireless', 12, 50);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 100, 'Bec ultra', 24, 20);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 100, 'Incarcator telefon', 16, 40);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 110, 'Lanterna', 36, 50);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 110, 'Incarcator tableta', NULL, 10);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 120, 'Aspirator robot', 10, 300);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 120, 'Fier de calcat', 12, 120);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 130, 'Uscator de par', 12, 90);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 130, 'Uscator de par Pro', 12, 190);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 140, 'Telefon', 14, 3000);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 140, 'Laptop', 14, 6000);
insert into produs (cod_produș, cod_furnizor, nume, luni_garanție, pret) values(seq_produș.nextval, 140, 'Tableta', 15, 1000);

select * from produs;

select *
from produs p, furnizor f
where p.cod_furnizor = f.cod_furnizor;

```

Script Output x Query Result x Query Result 1 x

SQL | All Rows Fetched: 12 in 0.024 seconds

	COD_PRODUS	COD_FURNIZOR	NUME	LUNI_GARANTIE	PRET
1	100	100	Casti Wireless	12	50
2	101	100	Bec ultra	24	20
3	102	100	Incarcator telefon	16	40
4	103	110	Lanterna	36	50
5	104	110	Incarcator tableta	(null)	10
6	105	120	Aspirator robot	10	300
7	106	120	Fier de calcat	12	120
8	107	130	Uscator de par	12	90
9	108	130	Uscator de par Pro	12	190
10	109	140	Telefon	14	3000
11	110	140	Laptop	14	6000
12	111	140	Tableta	15	1000

```

create sequence seq_pachet
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into pachet_promotional (cod_pachet, pret, puncte_fidelitate) values(seq_pachet.nextval, 70, 10);
insert into pachet_promotional (cod_pachet, pret, puncte_fidelitate) values(seq_pachet.nextval, 80, 12);
insert into pachet_promotional (cod_pachet, pret, puncte_fidelitate) values(seq_pachet.nextval, 6000, 30);
insert into pachet_promotional (cod_pachet, pret, puncte_fidelitate) values(seq_pachet.nextval, 420, 20);
insert into pachet_promotional (cod_pachet, pret, puncte_fidelitate) values(seq_pachet.nextval, 50, 3);

select * from pachet_promotional;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 5 in 0.022 seconds

	COD_PACHET	PRET	PUNCTE_FIDELITATE
1	100	70	10
2	110	80	12
3	120	6000	30
4	130	420	20
5	140	50	3

```

insert into contine (cod_produ, cod_pachet) values(100, 100);
insert into contine (cod_produ, cod_pachet) values(103, 100);
insert into contine (cod_produ, cod_pachet) values(104, 100);
insert into contine (cod_produ, cod_pachet) values(101, 110);
insert into contine (cod_produ, cod_pachet) values(107, 110);
insert into contine (cod_produ, cod_pachet) values(109, 120);
insert into contine (cod_produ, cod_pachet) values(110, 120);
insert into contine (cod_produ, cod_pachet) values(105, 120);
insert into contine (cod_produ, cod_pachet) values(105, 130);
insert into contine (cod_produ, cod_pachet) values(107, 130);
insert into contine (cod_produ, cod_pachet) values(103, 130);
insert into contine (cod_produ, cod_pachet) values(106, 130);
insert into contine (cod_produ, cod_pachet) values(111, 140);
insert into contine (cod_produ, cod_pachet) values(104, 140);
select * from contine;

```

Script Output x   Query Result x   Query Result 1 x   Query Result 2 x   Query Result 3 x			
SQL   All Rows Fetched: 14 in 0.013 seconds			
	COD_PRODUS	COD_PACHET	
1	100	100	
2	103	100	
3	104	100	
4	101	110	
5	107	110	
6	109	120	
7	110	120	
8	105	120	
9	105	130	
10	107	130	
11	103	130	
12	106	130	
13	111	140	

```

create sequence seq_utilizator
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola, data_autenticare) values(seq_utilizator.nextval, 100, 'marcel233', 'f123ds', to_date('12-05-2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola, data_autenticare) values(seq_utilizator.nextval, 110, 'marra4', '35323d', to_date('25-04-2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola, data_autenticare) values(seq_utilizator.nextval, 120, 'dvd3d4_a', 'parola', to_date('30-05-2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola, data_autenticare) values(seq_utilizator.nextval, 130, 'mira_12', 'prdfgd', to_date('12-04-2022', 'dd-mm-yyyy'));
insert into utilizator (cod_utilizator, cod_beneficiar, nume_utilizator, parola, data_autenticare) values(seq_utilizator.nextval, 150, 'mnt_ana', '567fg', to_date('02-05-2022', 'dd-mm-yyyy'));
select* from utilizator;

```

Script Output x   Query Result x   Query Result 1 x   Query Result 2 x   Query Result 3 x   Query Result 4 x				
SQL   All Rows Fetched: 5 in 0.024 seconds				
	COD_UTILIZATOR	COD_BENEFICIAR	NUME_UTILIZATOR	PAROLA DATA_AUTENTICARE
1	100	100	marcel233	f123ds 12-MAY-22
2	110	110	marra4	35323d 25-APR-22
3	120	120	dvd3d4_a	parola 30-MAY-22
4	130	130	mira_12	prdfgd 12-APR-22
5	140	150	mnt_ana	567fg 02-MAY-22

```

create sequence seq_premiu
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 100, 103, 60, 20);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 100, 100, 70, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 100, 108, 90, 40);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 110, 106, 80, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 110, 103, 90, 35);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 120, NULL, 70, 210);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 120, 102, 75, 20);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 130, 101, 80, NULL);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 130, 105, 99, 30);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 140, 111, 100, 0);
insert into premiu (cod_premiu, cod_concurs, cod_produs, punctaj_minim, suma) values (seq_premiu.nextval, 140, 104, 50, 10);
select* from premiu;

insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 110, 50);

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x

SQL All Rows Fetched: 11 in 0.026 seconds

	COD_PREMIU	COD_CONCURS	COD_PRODUS	PUNCTAJ_MINIM	SUMA
1	100	100	103	60	20
2	110	100	100	70	30
3	120	100	108	90	40
4	130	110	106	80	30
5	140	110	103	90	35
6	150	120	(null)	70	210
7	160	120	102	75	20
8	170	130	101	80	(null)
9	180	130	105	99	30
10	190	140	111	100	0
11	200	140	104	50	10

```

insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 100, 66);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 130, 74);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (100, 140, 95);

insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 100, 92);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 110, 94);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 130, 88);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (110, 140, 50);

insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 100, 97);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 110, 73);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (120, 140, 60);

insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 100, 99);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 110, 90);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 130, 79);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (130, 140, 73);

insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 100, 99);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 110, 45);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 130, 74);
insert into participa (cod_concurs, cod_utilizator, punctaj) values (140, 140, 100);
select* from participa;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

SQL | All Rows Fetched: 19 in 0.008 seconds

	COD_CONCURS	COD_UTILIZATOR	PUNCTAJ
1	100	110	50
2	100	100	66
3	100	130	74
4	100	140	95
5	110	100	92
6	110	110	94
7	110	130	88
8	110	140	50
9	120	100	97
10	120	110	73
11	120	140	60
12	130	100	99
13	130	110	90
14	130	130	79
15	130	140	73

```

create sequence seq_angajat
start with 100
increment by 10
maxvalue 1000
nocycle
nocache;

insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Lopataru', 'Alexandra', 3500, '+400758465748');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Cujbescu', 'Marius', 3600, '+400745678954');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Serban', 'Alina', 3700, '+400767584957');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Mihai', 'Costin', 3650, '+400778965435');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Lazar', 'Simon', 3700, '+400789657453');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Moldovan', 'Andreea', 5000, '+400709876547');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Oprea', 'Pavel', 5500, '+400798765467');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Dumitrescu', 'Bianca', 5600, '+400785768594');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Dinu', 'Dobrin', 5400, '+400734567895');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Cristea', 'Cristina', 5400, NULL);
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Todorescu', 'Simon', 6000, '+400734567845');
insert into angajat (cod_angajat, nume, prenume, salariu, telefon) values (seq_angajat.nextval, 'Anontescu', 'Briana', 7000, '+400787657893');
select * from angajat;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result 6 x Query Result 7 x Query Result 8 x

SQL | All Rows Fetched: 12 in 0.356 seconds

	COD_ANGAJAT	NUME	PRENUME	SALARIU	TELEFON
1	100	Lopataru	Alexandra	3500	+400758465748
2	110	Cujbescu	Marius	3600	+400745678954
3	120	Serban	Alina	3700	+400767584957
4	130	Mihai	Costin	3650	+400778965435
5	140	Lazar	Simon	3700	+400789657453
6	150	Moldovan	Andreea	5000	+400709876547
7	160	Oprea	Pavel	5500	+400798765467
8	170	Dumitrescu	Bianca	5600	+400785768594
9	180	Dinu	Dobrin	5400	+400734567895
10	190	Cristea	Cristina	5400	(null)
11	200	Todorescu	Simon	6000	+400734567845
12	210	Anontescu	Briana	7000	+400787657893

```

insert into scenarist (cod_angajat, zi_libera, experienta) values (100, 'miercuri', 6);
insert into scenarist (cod_angajat, zi_libera, experienta) values (110, 'luni', 8);
insert into scenarist (cod_angajat, zi_libera, experienta) values (120, 'luni', 10);
insert into scenarist (cod_angajat, zi_libera, experienta) values (130, 'joi', 7);
insert into scenarist (cod_angajat, zi_libera, experienta) values (140, 'vineri', 6);
select * from scenarist;

insert into administrator (cod_angajat, data_angajare) values (150, to_date('15-08-2021', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (160, to_date('12-09-2021', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (170, to_date('19-08-2022', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (180, to_date('02-11-2022', 'dd-mm-yyyy'));

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result 6 x

SQL | All Rows Fetched: 5 in 0.015 seconds

	COD_ANGAJAT	ZI_LIBERA	EXPERIENTA
1	100	miercuri	6
2	110	luni	8
3	120	luni	10
4	130	joi	7
5	140	vineri	6



```

insert into administrator (cod_angajat, data_angajare) values (150, to_date('15-08-2021', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (160, to_date('12-09-2021', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (170, to_date('19-08-2022', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (180, to_date('02-11-2022', 'dd-mm-yyyy'));
insert into administrator (cod_angajat, data_angajare) values (190, to_date('16-05-2023', 'dd-mm-yyyy'));
select* from administrator;

```

create sequence seq\_reclama

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result

All Rows Fetched: 5 in 0.018 seconds

	COD_ANGAJAT	DATA_ANGAJARE
1	150	15-AUG-21
2	160	12-SEP-21
3	170	19-AUG-22
4	180	02-NOV-22
5	190	16-MAY-23

```

create sequence seq_reclama
start with 100
increment by 10
maxvalue 1000
nocycle;

insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 100, 140, 30, 2150);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 100, 130, 40, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, NULL, 130, 40, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 101, 110, 50, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 101, 130, 60, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 101, 140, 10, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 102, 110, 20, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 102, 140, 30, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 102, 130, 35, 2250);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 103, 120, 35, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 103, 110, 35, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 105, 120, 30, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 105, 140, 45, 1400);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 106, 130, 24, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 106, 120, 45, 3000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 107, 110, 30, 1220);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 109, 140, 30, 500);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 110, 130, 45, 4000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 111, 110, 55, 5000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 110, 120, 30, 4030);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, NULL, 120, 42, 2000);
insert into reclama (cod_reclama, cod_produs, cod_angajat, durata, cost_total) values (seq_reclama.nextval, 108, 130, 25, 600);
select * from reclama;

```

All Rows Fetched: 22 in 0.016 seconds

	COD_RECLAMA	COD_PRODUS	COD_ANGAJAT	DURATA	COST_TOTAL
1	100	100	140	30	2150
2	110	100	130	40	2250
3	120	(null)	130	40	1400
4	130	101	110	50	1400
5	140	101	130	60	2250
6	150	101	140	10	3000
7	160	102	110	20	3000
8	170	102	140	30	2250
9	180	102	130	35	2250
10	190	103	120	35	1220

```

insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values (seq_sediu.nextval, 150, 'Electric center', 'Bucuresti Sud', 10000, 'electric.center@gmail.com');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values (seq_sediu.nextval, 160, 'Cable power', 'Centru Bucuresti City', 12000, 'cable.power234@yahoo.ro');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values (seq_sediu.nextval, 170, 'Atacul tehnologiei', 'Ploiesti Periferie', 5000, 'technology.attack578@yahoo.com');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values (seq_sediu.nextval, 170, 'IT Party', 'Buzau Cartier Dorobanti', 6000, 'partyit45@gmail.ro');
insert into sediu (cod_sediu, cod_angajat, nume, adresa, suprafata, adresa_mail) values (seq_sediu.nextval, 190, 'ELECTRO LIGHT MAIN', 'Bucuresti Strada Pacii', 12000, 'electrowizBucharest@gmail.com');
select* from sediu;

insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 110);

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result 6 x Query Result 7 x Query Result 8 x Query Result 9 x Query Result 10 x Query Result 11

All Rows Fetched: 5 in 0.02 seconds

	COD_SEDIU	COD_ANGAJAT	NUME	ADRESA	SUPRAFATA	ADRESA_MAIL
1	100	150	Electric center	Bucuresti Sud	10000	electric.center@gmail.com
2	110	160	Cable power	Centru Bucuresti City	12000	cable.power234@yahoo.ro
3	120	170	Atacul tehnologiei	Ploiesti Periferie	5000	technology.attack578@yahoo.com
4	130	170	IT Party	Buzau Cartier Dorobanti	6000	partyit45@gmail.ro
5	140	190	ELECTRO LIGHT MAIN	Bucuresti Strada Pacii	12000	electrowizBucharest@gmail.com



Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

```

insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(100, 130);

insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 100);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(110, 140);

insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(120, 140);

insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 120);
insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 130);
insert into alimenteaza (cod_furnizor, cod_sediu) values(130, 140);

insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 100);
insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 110);
insert into alimenteaza (cod_furnizor, cod_sediu) values(140, 140);

```

SQL | All Rows Fetched: 16 in 0.016 seconds

	COD_FURNIZOR	COD_SEDIU
1	100	110
2	100	120
3	100	130
4	110	100
5	110	110
6	110	120
7	110	140
8	120	110
9	120	120
10	120	140
11	130	120
12	130	130
13	130	140
14	140	100
15	140	110
16	140	140

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result 5 x Query Result 6 x Query Result 7 x Query Result 8

```

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (100, 100, 120, to_date('17-08-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (100, 130, 120, to_date('15-10-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (100, 140, 100, to_date('14-11-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (100, 120, 140, to_date('12-12-2022', 'dd-mm-yyyy'));

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 100, 120, to_date('13-08-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 110, 120, to_date('16-07-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 110, 130, to_date('17-05-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 120, 140, to_date('18-03-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 130, 120, to_date('19-02-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (110, 140, 110, to_date('20-01-2020', 'dd-mm-yyyy'));

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (120, 110, 120, to_date('21-11-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (120, 120, 140, to_date('22-12-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (120, 140, 100, to_date('23-04-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (120, 130, 120, to_date('24-04-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (120, 100, 110, to_date('25-06-2021', 'dd-mm-yyyy'));

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (130, 100, 120, to_date('26-06-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (130, 120, 140, to_date('27-07-2022', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (130, 110, 130, to_date('28-09-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (130, 140, 110, to_date('29-08-2020', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (130, 110, 120, to_date('30-10-2019', 'dd-mm-yyyy'));

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (140, 110, 120, to_date('25-03-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (140, 130, 120, to_date('01-03-2019', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (140, 120, 140, to_date('02-09-2021', 'dd-mm-yyyy'));

insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (150, 100, 110, to_date('02-06-2018', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (150, 110, 120, to_date('03-06-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (150, 120, 110, to_date('04-07-2018', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (150, 120, 140, to_date('05-07-2021', 'dd-mm-yyyy'));
insert into achizitioneaza (cod_beneficiar, cod_pachet, cod_sediu,data_achizitionare) values (150, 130, 120, to_date('06-08-2019', 'dd-mm-yyyy'));

```

SQL | All Rows Fetched: 30 in 0.016 seconds

	COD_BENEFICIAR	COD_PACHET	COD_SEDIU	DATA_ACHIZITIONARE
1	100	100	110	16-SEP-22
2	100	100	120	17-AUG-22
3	100	130	120	15-OCT-21
4	100	140	100	14-NOV-21
5	100	120	140	12-DEC-22
6	110	100	120	13-AUG-21
7	110	110	110	16-JUL-22

## 12) INTEROGARI EXERCITIUL 12:

--Interogare 1: afișați codul beneficiarilor în ordinea punctelor de fidelitate obținute din pachete cumpărate de la sediul/sediile cu cele mai multe produse, precizându-se și suma acestor puncte

--ordonati descrescator dupa totalul punctelor de fidelitate

--sa se afiseze doar beneficiarii cu cel putin 4 puncte de fidelitate acumulate

--elemente folosite: subcerere corelata in care intervin cel putin 3 tabele,

--grupari de date cu subcereri nesincronizate (subcererea necorelata pe 4 tabele din interogare se afla in clauza from a altei subcereri) in care intervin cel putin 3 tabele, functii grup(count, sum si max), filtrare la nivel de grupuri(in cadrul aceleiasi cereri)

```
select b.cod_beneficiar, sum(p.puncte_fidelitate) total_puncte
from beneficiar b, achizitioneaza a, pachet_promotional p
where a.cod_beneficiar = b.cod_beneficiar and p.cod_pachet = a.cod_pachet
and
    (select count(*) --subcerere corelata, calculeaza nr de produse din sediul curent
     from alimenteaza a1, furnizor f, produs p1
     where a1.cod_sediu = a.cod_sediu and a1.cod_furnizor = f.cod_furnizor and
          p1.cod_furnizor = f.cod_furnizor)
=
    (select max(nr_produce) from( --subcerere necorelata cu 4 tabele in clauza from a
    unei subcereri, calculeaza numarul de produse per sediu, si apoi maximul
    select s.cod_sediu as cod, count(*) as nr_produce
    from sediu s, alimenteaza a, furnizor f, produs p
    where a.cod_sediu = s.cod_sediu and a.cod_furnizor = f.cod_furnizor and
          p.cod_furnizor = f.cod_furnizor
    group by s.cod_sediu))
group by b.cod_beneficiar
having sum(p.puncte_fidelitate) >= 4 --filtrare la nivel de grupuri
order by total_puncte desc;
```

--Interogare 1: afisati codul beneficiarilor in ordinea punctelor de fidelitate obtinute din pachete cumparate de la sediul/sediile cu cele mai multe produse, precizandu-se si suma acestor puncte

--ordonati descrescator dupa totalul punctelor de fidelitate

--sa se afiseze doar beneficiarii cu cel putin 4 puncte de fidelitate acumulate

--elemente folosite: subcerere corelata in care intervin cel putin 3 tabele,

--grupari de date cu subcereri nesincronizate (subcererea necorelata pe 4 tabele din interogare se afla in clauza from a altei subcereri) in care intervin cel putin 3 tabele, functii grup, filtrare la nivel de grupuri

```
select b.cod_beneficiar, sum(p.puncte_fidelitate) total_puncte
from beneficiar b, achizitioneaza a, pachet_promotional p
where a.cod_beneficiar = b.cod_beneficiar and p.cod_pachet = a.cod_pachet
and
    (select count(*) --subcerere corelata, calculeaza nr de produse din sediul curent
     from alimenteaza a1, furnizor f, produs p1
     where a1.cod_sediu = a.cod_sediu and a1.cod_furnizor = f.cod_furnizor and p1.cod_furnizor = f.cod_furnizor)
=
    (select max(nr_produce) from( --subcerere necorelata cu 4 tabele in clauza from a unei subcereri, calculeaza maximul de produse dintr-un sediu, in cazul nostru sediul cu nr maxim este 110, avand 10 produse
    select s.cod_sediu as cod, count(*) as nr_produce
    from sediu s, alimenteaza a, furnizor f, produs p
    where a.cod_sediu = s.cod_sediu and a.cod_furnizor = f.cod_furnizor and p.cod_furnizor = f.cod_furnizor
    group by s.cod_sediu))
group by b.cod_beneficiar
having sum(p.puncte_fidelitate) >= 4
order by total_puncte desc;
```

Query Result 1 x Query Result 2 x Query Result 3 x Script Output x Query Result 4 x Query Result 5 x

All Rows Fetched: 3 in 0.054 seconds

	COD_BENEFICIAR	TOTAL_PUNCTE
1	150	43
2	100	10
3	120	10

**--Interogare 2:** s-a stabilit creșterea perioadei de garanție pentru produsele cu cea mai scurtă garanție, astfel că produselor cu 0 garanție li se vor aduna 6 luni de garanție, celor cu 10 4 luni, celor cu 12 2 luni, iar celelalte vor rămâne la fel

--să se afișeze pentru fiecare produs (codul și numele) și numărul de premii în care e oferit în cadrul concursurilor din ultimele 100 de zile  
 --afișați și garanția inițială și cea finală  
 --ordonați crescător după valoarea noii garanții și după numărul de premii  
 --**elemente folosite:** subcerere corelată cu 3 tabele,  
 -- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)  
 --o funcție pe data calendaristică(sysdate)

```
select cod_produs, nume, nvl(luni_garanție, 0) as garanție_initiala, decode
(nvl(luni_garanție, 0), 0, 4, 10, luni_garanție + 4, 12, 14, luni_garanție) as noua_garanție,
(select count(*) -- subcerere corelată în clauza select, determină în câte premii recente
apare produsul
  from(select *
        from premiu p, concurs c
        where p.cod_concurs = c.cod_concurs and pr.cod_produs = p.cod_produs and
        (sysdate - c.data_concurs) <= 100 )
) as numar_premii
from produs pr
order by noua_garanție, numar_premii;
```

--Interogare 2: s-a stabilit creșterea perioadei de garanție pentru produsele cu cea mai scurtă garanție, astfel ca produselor cu 0 garanție li se vor aduna 6 luni de garanție, celor cu 10 4 luni, celor cu 12 2 luni, iar celelalte vor rămâne la fel

--sa se afișeze pentru fiecare produs (codul și numele) și numărul de premii în care e oferit în cadrul concursurilor din ultimele 100 de zile

--elemente folosite: subcerere corelată cu 3 tabele,  
 -- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

```
select cod_produs, nume, nvl(luni_garanție, 0) as garanție_initiala, decode (nvl(luni_garanție, 0), 0, 4, 10, luni_garanție + 4, 12, 14, luni_garanție) as noua_garanție,
(select count(*) -- subcerere corelată în clauza select, determină în câte premii recente apare produsul
  from(select *
        from premiu p, concurs c
        where p.cod_concurs = c.cod_concurs and pr.cod_produs = p.cod_produs and (sysdate - c.data_concurs) <= 100 )
) as numar_premii
from produs pr
order by noua_garanție, numar_premii;
```

-- Interogare 3: sa se determine care sunt beneficiarii care isi sarbatoresc ziua primavara si care au participat la toate concursurile desfasurate in anul 2023 si care au

with concursuri\_2023\_peste\_10\_intrebări as (select cod\_concurs as cod --stochează concursurile care respectă condițiile

	COD_PRODUS	NUME	GARANȚIE_INITIALA	NOUA_GARANȚIE	NUMAR_PREMII
1	104	Incarcator tableta	0	4	0
2	105	Aspirator robot	10	14	0
3	110	Laptop	14	14	0
4	109	Telefon	14	14	0
5	107	Uscator de par	12	14	0
6	106	Fier de calcat	12	14	1
7	108	Uscator de par Pro	12	14	1
8	100	Casti Wireless	12	14	1
9	111	Tableta	15	15	0
10	102	Incarcator telefon	16	16	1
11	101	Bec ultra	24	24	0
12	103	Lanterna	36	36	2

-- **Interogare 3:** să se determine care sunt beneficiarii (nume și prenume concatenate) care își sărbătoresc ziua primăvară și care au participat la toate concursurile desfășurate în anul 2023 și care au peste 10 întrebări

--**elemente folosite:** un bloc de cerere, subcerere nesincronizată în clauza FROM, o funcție pe dată calendaristică (months\_between)

-- 2 funcții pe șiruri de caractere ( concat și substr)

```
with concursuri_2023_peste_10_intrebări as (select cod_concurs as cod --stocheaza
concursurile care respecta conditiile
    from concurs
    where MONTHS_BETWEEN(to_date('01-01-2024', 'dd-mm-yyyy'), data_concurs)
> 0 and MONTHS_BETWEEN(to_date('01-01-2024', 'dd-mm-yyyy'), data_concurs) <=
12 and nr_intrebări > 10),
numar_concursuri as(select count(*) nr_concursuri --calculeaza cate concursuri sunt
    from concursuri_2023_peste_10_intrebări)
select nume_beneficiar
from (select b.cod_beneficiar, concat(concat(b.num, ' '), b.prenume) as
nume_beneficiar, count(*) as nr_conc --subcerere necorelata in clauza from, stocheaza
la cate concursuri care respecta conditia a participat fiecare beneficiar
    from participa p, beneficiar b, utilizator u
    where cod_concurs in (select cod from concursuri_2023_peste_10_intrebări) and
p.cod_utilizator = u.cod_utilizator and u.cod_beneficiar = b.cod_beneficiar
and substr((to_char(b.CNP)), 4, 2) in ('03', '04', '05')
    group by b.cod_beneficiar, concat(concat(b.num, ' '), b.prenume))
where (select nr_concursuri from numar_concursuri) = nr_conc;
```

```
-- Interogare 3: sa se determine care sunt beneficiarii (nume si prenume concatenate) care isi sarbatoresc ziua primavara si care au participat la toate concursurile desfasurate in an
--elemente folosite: un bloc de cerere, subcerere nesincronizata in clauza FROM, o functie pe data calendaristica,
-- 2 functii pe siruri de caractere ( concat si substr)

with concursuri_2023_peste_10_intrebări as (select cod_concurs as cod --stocheaza concursurile care respecta conditiile
    from concurs
    where MONTHS_BETWEEN(to_date('01-01-2024', 'dd-mm-yyyy'), data_concurs) > 0 and MONTHS_BETWEEN(to_date('01-01-2024', 'dd-mm-yyyy'), data_concurs) <= 12 and nr_intrebări > 10),
numar_concursuri as(select count(*) nr_concursuri --calculeaza cate concursuri sunt
    from concursuri_2023_peste_10_intrebări)
select nume_beneficiar
from (select b.cod_beneficiar, concat(concat(b.num, ' '), b.prenume) as nume_beneficiar, count(*) as nr_conc --subcerere necorelata in clauza from, stocheaza la cate concursuri care
    from participa p, beneficiar b, utilizator u
    where cod_concurs in (select cod from concursuri_2023_peste_10_intrebări) and p.cod_utilizator = u.cod_utilizator and u.cod_beneficiar = b.cod_beneficiar
and substr((to_char(b.CNP)), 4, 2) in ('03', '04', '05')
    group by b.cod_beneficiar, concat(concat(b.num, ' '), b.prenume))
where (select nr_concursuri from numar_concursuri) = nr_conc;
```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 2 in 0.048 seconds

NUME_BENEFICIAR
1 Suditu Mara
2 Munteanu Ana

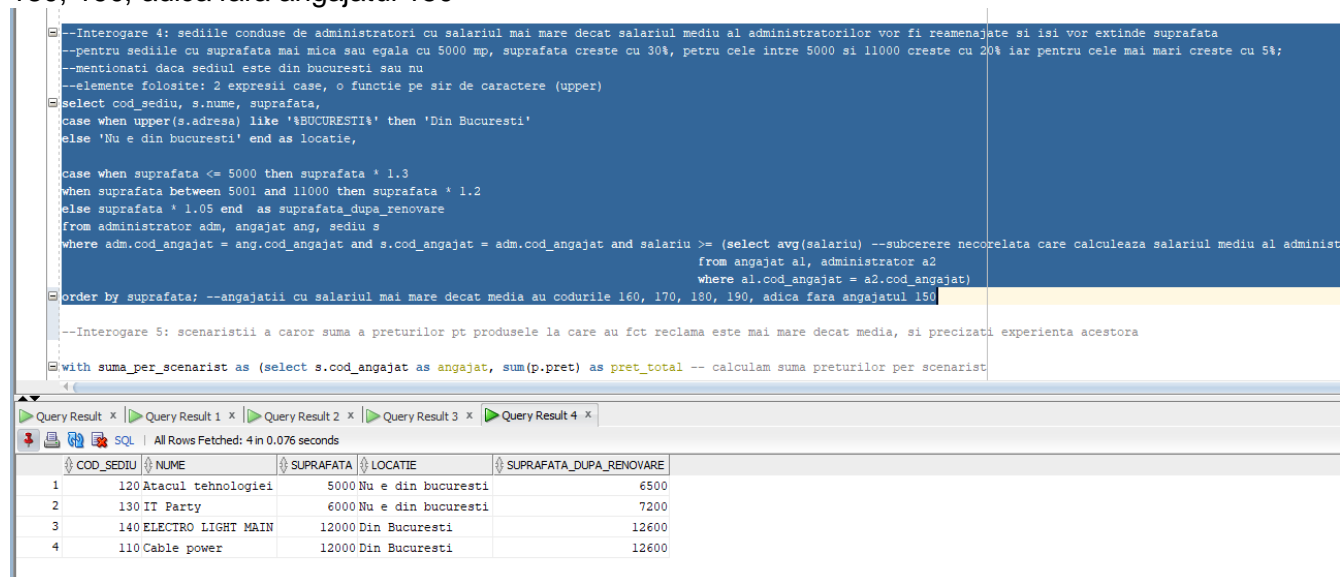
**--Interogare 4:** sediile conduse de administratori cu salariul mai mare decât salariul mediu al administratorilor vor fi reamenajate și își vor extinde suprafața  
--pentru sediile cu suprafața mai mică sau egală cu 5000 mp, suprafața crește cu 30%, pentru cele între 5000 și 11000 crește cu 20% iar pentru cele mai mari crește cu 5%;  
menționați dacă sediul este din bucurești sau nu și afișați codul, numele, suprafața anterioară și cea după renovare  
--ordonați după vechea suprafață  
**--elemente folosite:** 2 expresii case, o funcție pe șir de caractere (upper)

```
select cod_sediu, s.nume, suprafata,
case when upper(s.adresa) like '%BUCURESTI%' then 'Din Bucuresti'
else 'Nu e din bucuresti' end as locatie,
```

```
case when suprafata <= 5000 then suprafata * 1.3
when suprafata between 5001 and 11000 then suprafata * 1.2
else suprafata * 1.05 end as suprafata_dupa_renovare
from administrator adm, angajat ang, sediu s
where adm.cod_angajat = ang.cod_angajat and s.cod_angajat = adm.cod_angajat and
salariu >= (select avg(salariu) --subcerere necorelata care calculeaza salariul mediu al
administratorilor
```

```
from angajat a1, administrator a2
where a1.cod_angajat = a2.cod_angajat)
```

```
order by suprafata; --angajatii cu salariul mai mare decat media au codurile 160, 170,
180, 190, adica fara angajatul 150
```



The screenshot shows a SQL query editor with the following code:

```
--Interogare 4: sediile conduse de administratori cu salariul mai mare decat salariul mediu al administratorilor vor fi reamenajate si isi vor extinde suprafata
--pentru sediile cu suprafata mai mica sau egala cu 5000 mp, suprafata creste cu 30%, pentru cele intre 5000 si 11000 creste cu 20% iar pentru cele mai mari creste cu 5%;
--mentionati daca sediul este din bucuresti sau nu
--elemente folosite: 2 expresii case, o functie pe sir de caractere (upper)
select cod_sediu, s.nume, suprafata,
case when upper(s.adresa) like '%BUCURESTI%' then 'Din Bucuresti'
else 'Nu e din bucuresti' end as locatie,

case when suprafata <= 5000 then suprafata * 1.3
when suprafata between 5001 and 11000 then suprafata * 1.2
else suprafata * 1.05 end as suprafata_dupa_renovare
from administrator adm, angajat ang, sediu s
where adm.cod_angajat = ang.cod_angajat and s.cod_angajat = adm.cod_angajat and salariu >= (select avg(salariu) --subcerere necorelata care calculeaza salariul mediu al administ
from angajat a1, administrator a2
where a1.cod_angajat = a2.cod_angajat)

order by suprafata; --angajatii cu salariul mai mare decat media au codurile 160, 170, 180, 190, adica fara angajatul 150

--Interogare 5: scenaristii a caror suma a preturilor pt produsele la care au fct reclama este mai mare decat media, si precizata experienta acestora

with suma_per_scenarist as (select s.cod_angajat as angajat, sum(p.pret) as pret_total -- calculam suma preturilor per scenarist
```

The results are displayed in a table with the following columns: COD\_SEDIU, NUME, SUPRAFATA, LOCATIE, and SUPRAFATA\_DUPA\_RENOVARE.

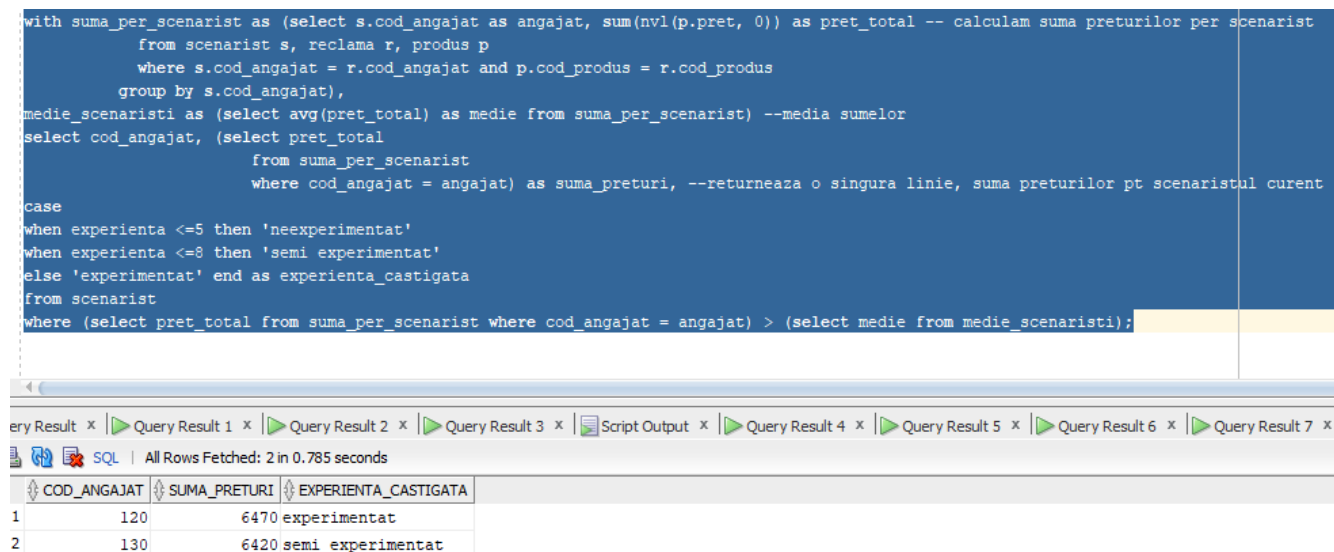
COD_SEDIU	NUME	SUPRAFATA	LOCATIE	SUPRAFATA_DUPA_RENOVARE
1	120 Atacul tehnologiei	5000	Nu e din bucuresti	6500
2	130 IT Party	6000	Nu e din bucuresti	7200
3	140 ELECTRO LIGHT MAIN	12000	Din Bucuresti	12600
4	110 Cable power	12000	Din Bucuresti	12600

**--Interogare 5:** afișați scenariștii (codul) a căror sumă a prețurilor pt produsele la care au făcut reclamă este mai mare decât media, precizând și media prețurilor pentru fiecare scenarist

--se va preciza și experiența acestora, menționând dacă este neexperimentat (experiență mai mică de 5), semi experimentat sau experimentat (experiență peste 7 puncte)

**--elemente folosite:** bloc de cerere, o expresie case

```
with suma_per_scenarist as (select s.cod_angajat as angajat, sum(nvl(p.pret, 0)) as pret_total -- calculam suma preturilor per scenarist
    from scenarist s, reclama r, produs p
    where s.cod_angajat = r.cod_angajat and p.cod_produc = r.cod_produc
    group by s.cod_angajat),
medie_scenaristi as (select avg(pret_total) as medie from suma_per_scenarist) --media sumelor
select cod_angajat, (select pret_total
    from suma_per_scenarist
    where cod_angajat = angajat) as suma_preturi, --returneaza o singura
linie, suma preturilor pt scenaristul curent
case
when experienta <=5 then 'neexperimentat'
when experienta <=8 then 'semi experimentat'
else 'experimentat' end as experienta_castigata
from scenarist
where (select pret_total from suma_per_scenarist where cod_angajat = angajat) >
(select medie from medie_scenaristi);
```



```
with suma_per_scenarist as (select s.cod_angajat as angajat, sum(nvl(p.pret, 0)) as pret_total -- calculam suma preturilor per scenarist
    from scenarist s, reclama r, produs p
    where s.cod_angajat = r.cod_angajat and p.cod_produc = r.cod_produc
    group by s.cod_angajat),
medie_scenaristi as (select avg(pret_total) as medie from suma_per_scenarist) --media sumelor
select cod_angajat, (select pret_total
    from suma_per_scenarist
    where cod_angajat = angajat) as suma_preturi, --returneaza o singura linie, suma preturilor pt scenaristul curent
case
when experienta <=5 then 'neexperimentat'
when experienta <=8 then 'semi experimentat'
else 'experimentat' end as experienta_castigata
from scenarist
where (select pret_total from suma_per_scenarist where cod_angajat = angajat) > (select medie from medie_scenaristi);
```

COD_ANGAJAT	SUMA_PRETURI	EXPERIENTA_CASTIGATA
1	120	6470 experimentat
2	130	6420 semi experimentat

### 13) EXERCITIUL 13:

#### --EXERCITIUL 13: 3 operații de update

*--să se mărească cu 10% prețul produselor care apar în măcar un premiu și care au prețul mai mic decât media produselor de la același furnizor*

```
update produs
set pret = pret * 1.1
where pret < (select avg(pret)
              from produs p
              where cod_furnizor = p.cod_furnizor)
and cod_produș in (select pr.cod_produș
                  from premiu pr)
and cod_furnizor is not null;
```

*--să se adauge 2 întrebări (să se crească numărul de întrebări cu 2) la concursurile ale căror premii ( considerate o singură dată) în bani valorează împreună peste 100 de lei*

*--tot la aceste concursuri să se crească taxa de înscriere cu minimul taxei de înscriere dintre toate concursurile*

```
update concurs
set nr_intrebari = nr_intrebari + 2, taxa_inscriere = taxa_inscriere + (select
min(cn.taxa_inscriere)
                                from concurs cn)
where 100 < (select sum(nvl(p.suma, 0))
            from premiu p
            where p.cod_concurs = cod_concurs);
```

*--să se crească cu 10% prețul pachetelor promoționale care conțin cel puțin un produs de la furnizorul cu numele 'IT DIRECT'*

```
update pachet_promotional
set pret = pret * 1.1
where cod_pachet in (select p.cod_pachet
                    from pachet_promotional p join contine c on(p.cod_pachet = c.cod_pachet)
                    join produs pr on(c.cod_produș = pr.cod_produș) join furnizor f on(pr.cod_furnizor =
f.cod_furnizor)
                    where f.num = 'IT Direct'
                    group by p.cod_pachet);
```

*--operație de șuprimare:*

*--să se șteargă vânzările din anul 2019 care au avut loc în sediul/sediile cu cea mai mică suprafață*

```

delete from achizitioneaza
where cod_sediu in (select s.cod_sediu
                    from sediu s
                    where s.suprafata = (select min(ss.suprafata) from sediu ss) )
and to_char(data_achizitionare, 'YYYY') = '2019';

```

```

--EXERCITIUL 13: 3 operatii de update

--sa se mareasca cu 10% pretul produselor care apar in macar un premiu si care au pretul mai mic decat media produselor de la acelasi furnizor
update produs
set pret = pret * 1.1
where pret < (select avg(pret)
              from produs p
              where cod_furnizor = p.cod_furnizor)
and cod_produs in (select pr.cod_produs
                  from premiu pr)
and cod_furnizor is not null;

```

Query Result x Script Output x

Task completed in 0.2 seconds

8 rows updated.

```

update concurs
set nr_intrebari = nr_intrebari + 2, taxa_inscriere = taxa_inscriere + (select min(cn.taxa_inscriere)
                                                                    from concurs cn)
where 100 < (select sum(nvl(p.suma, 0))
            from premiu p
            where p.cod_concurs = cod_concurs);

```

Query Result x Script Output x

Task completed in 0.123 seconds

5 rows updated.

```

--sa se creasca cu 10% pretul pachetelor promotionale care contin cel putin un produs de la furnizorul cu numele 'IT DIRECT'
update pachet_promotional
set pret = pret * 1.1
where cod_pachet in (select p.cod_pachet
                    from pachet_promotional p join contine c on (p.cod_pachet = c.cod_pachet) join produs pr on (c.cod_produs = pr.cod_produs) join furnizor f on (pr.cod_furnizor = f.cod_furnizor)
                    where f.nume = 'IT Direct'
                    group by p.cod_pachet);

--operatie de suprimare:
--sa se stearga vanzarile din anul 2019 care au avut loc in sediul/sediile cu cea mai mica suprafata
delete from achizitioneaza

```

Query Result x Script Output x

Task completed in 0.197 seconds

2 rows updated.

```

--operatie de suprimare:
--sa se stearga vanzarile din anul 2019 care au avut loc in sediul/sediile cu cea mai mica suprafata
delete from achizitioneaza
where cod_sediu in (select s.cod_sediu
                    from sediu s
                    where s.suprafata = (select min(ss.suprafata) from sediu ss) )
and to_char(data_achizitionare, 'YYYY') = '2019';

```

Query Result x Query Result 1 x Script Output x

Task completed in 0.109 seconds

3 rows deleted.



## 14) EXERCITIUL 15:

### --Analiza TOP n:

--să se afișeze primii 3 utilizatori (cod utilizator) în ordinea sumei câștigate din premii în bani în cadrul concursurilor la care au participat

--să se precizeze și suma câștigată pentru fiecare utilizator

mențiune: un concurent câștigă premiul din cadrul concursului cu punctajul minim necesar cel mai apropiat de punctajul obținut de concurent, dar nu mai mare

```
select *
from(select subcerere.*, rownum indice
      from
      (
        select ut as cod_user, sum(nvl(bani_premiu, 0)) as total_bani from
          (select u.cod_utilizator as ut, pr.cod_premiu, c.cod_concurs, pr.suma as bani_premiu
            from utilizator u join participa p on(u.cod_utilizator = p.cod_utilizator) join concurs c
            on(p.cod_concurs = c.cod_concurs) join premiu pr on(c.cod_concurs = pr.cod_concurs)
          where p.punctaj > pr.punctaj_minim
          and p.punctaj - pr.punctaj_minim = (select min(p.punctaj - pp.punctaj_minim)
                                             from concurs cc join premiu pp
                                             on(cc.cod_concurs = pp.cod_concurs)
          where cc.cod_concurs = c.cod_concurs
          and p.punctaj > pp.punctaj_minim))
        group by ut
        order by total_bani desc
      ) subcerere
    )
where indice <=3;
```

```
--Analiza TOP n:
--sa se afiseze primii 3 utilizatori (cod utilizator) in ordinea sumei castigate din premii in bani in cadrul concursurilor la care au participat
--sa se precizeze si suma castigata pentru fiecare utilizator
--mentione: un concurent castiga premiul din cadrul concursului cu punctajul minim necesar cel mai apropiat de punctajul obtinut de concurent, dar nu mai mare

select *
from (select subcerere.i, rownum indice
      from
      (
        select ut as cod_user, sum(nvl(bani_premiu, 0)) as total_bani from
        (select u.cod_utilizator as ut, pr.cod_premiu, c.cod_concurs, pr.suma as bani_premiu
         from utilizator u join participa p on(u.cod_utilizator = p.cod_utilizator) join concurs c on(p.cod_concurs = c.cod_concurs) join premiu pr on(c.cod_concurs = pr.cod_concurs)
         where p.punctaj > pr.punctaj_minim
         and p.punctaj - pr.punctaj_minim = (select min(p.punctaj - pp.punctaj_minim
                                         from concurs cc join premiu pp on(cc.cod_concurs = pp.cod_concurs)
                                         where cc.cod_concurs = c.cod_concurs and p.punctaj > pp.punctaj_minim))
        group by ut
        order by total_bani desc
      ) subcerere
      where indice <=3;

```

COD_USER	TOTAL_BANI	INDICE
110	245	1
100	85	2
130	70	3

## --OPERATIA DIVISION, METODA CU 2 NOT EXISTS:

--sa se afiseze codurile si numele furnizorilor care alimenteaza toate sediile din Bucuresti

--sa se afiseze si numarul de produse furnizate de fiecare dintre acesti furnizori si sa se ordoneze descrescator dupa acest numar

select f.cod\_furnizor as cod, f.numa as nume\_furnizor, count(\*) as numar\_produce\_furnizate

from alimenteaza a join furnizor f on (a.cod\_furnizor = f.cod\_furnizor) join produs p  
on(f.cod\_furnizor = p.cod\_furnizor)

where not exists

(select 'a'

from sediu s

where adresa like '%Bucuresti%'

and not exists

(select 'b'

from alimenteaza a1

where a.cod\_furnizor = a1.cod\_furnizor and s.cod\_sediu = a1.cod\_sediu

))

group by f.cod\_furnizor, f.numa

order by numar\_produce\_furnizate desc;

```
--OPERATIA DIVISION, METODA CU 2 NOT EXISTS:
--sa se afiseze codurile si numele furnizorilor care alimenteaza toate sediile din bucuresti
--sa se afiseze si numarul de produse furnizate de fiecare dintre acesti furnizori si sa se ordoneze descrescator dupa acest numar
select f.cod_furnizor as cod, f.numa as nume_furnizor, count(*) as numar_produce_furnizate
from alimenteaza a join furnizor f on (a.cod_furnizor = f.cod_furnizor) join produs p on (f.cod_furnizor = p.cod_furnizor)
where not exists
(
select 'a'
from sediu s
where adresa like '%Bucuresti%'
and not exists
(
select 'b'
from alimenteaza al
where a.cod_furnizor = al.cod_furnizor and s.cod_sediu = al.cod_sediu
))
group by f.cod_furnizor, f.numa
order by numar_produce_furnizate desc;
```

COD	NUME_FURNIZOR	NUMAR_PRODUSE_FURNIZATE
1	140 Ultra circuite	9
2	110 Electrtic party	8

## --OUTER JOIN PE 4 TABELE:

--afisati pentru fiecare beneficiar codul, numele si prenumele concatenate, numele de utilizator si data celui mai recent concurs la care a participat

--se va ordona dupa aceasta data

--daca nu a participat la niciun concurs se va afisa "nu a participat la concursuri" in dreptul coloanei aferente datei

--daca beneficiarul nu are nici utilizator (si implicit nu a participat nici la concursuri), se va specifica "nu are cont de utilizator" in locul numelui de utilizator

```
select b.cod_beneficiar, concat(concat(b.numa, ' '), b.prenume) as nume, nvl(u.numa_utilizator, 'nu are cont de utiliator') as date_utilizator, nvl(to_char(max(c.data_concurs), 'dd-mm-yyyy'), 'nu a participat la concursuri') as cea_mai_recenta_data
```

```
from beneficiar b, utilizator u, participa p, concurs c
```

```
where b.cod_beneficiar = u.cod_beneficiar(+) and u.cod_utilizator = p.cod_utilizator(+) and p.cod_concurs = c.cod_concurs(+)
```

```
group by b.cod_beneficiar, concat(concat(b.numa, ' '), b.prenume), nvl(u.numa_utilizator, 'nu are cont de utiliator')
```

```
order by cea_mai_recenta_data;
```

```
select b.cod_beneficiar, concat(concat(b.numa, ' '), b.prenume) as nume, nvl(u.numa_utilizator, 'nu are cont de utiliator') as date_utilizator, nvl(to_char(max(c.data_concurs), 'dd-mm-yyyy'), 'nu a participat la concursuri') as cea_mai_recenta_data
from beneficiar b, utilizator u, participa p, concurs c
where b.cod_beneficiar = u.cod_beneficiar(+) and u.cod_utilizator = p.cod_utilizator(+) and p.cod_concurs = c.cod_concurs(+)
group by b.cod_beneficiar, concat(concat(b.numa, ' '), b.prenume), nvl(u.numa_utilizator, 'nu are cont de utiliator')
order by cea_mai_recenta_data;
```

COD_BENEFICIAR	NUME	DATE_UTILIZATOR	CEA_MAI_RECENTA_DATA
1	110 Suditu Mara	marra4	18-04-2023
2	100 Andronic Marcel	marcel233	18-04-2023
3	130 Lupu Eugen	mira_12	18-04-2023
4	150 Munteanu Ana	mnt_ana	18-04-2023
5	120 Antonescu David	dvd34_a	nu a participat la concursuri
6	140 Alexandru Miruna	nu are cont de utilizator	nu a participat la concursuri

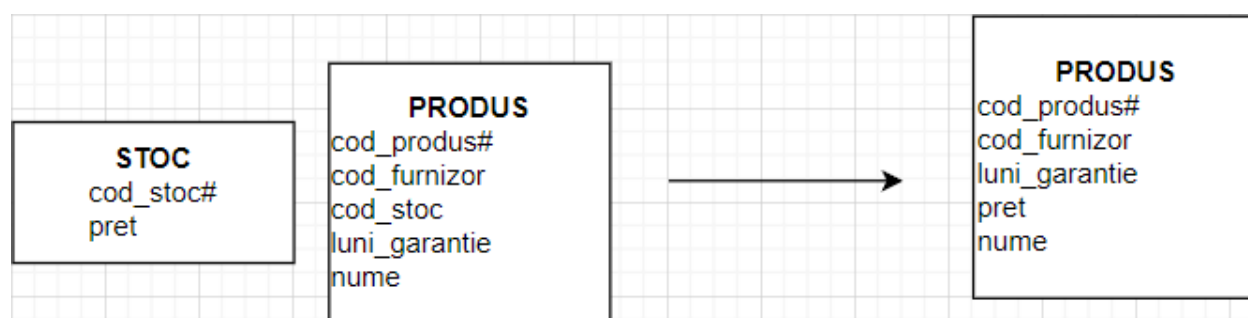
## 15)DENORMALIZAREA SI JUSTIFICAREA IMPORTANTEI ACESTEIA:

Presupunem ca mai există o relatie **STOC** (cod\_stoc#, pret), a carei cheie primara cod\_stoc devine cheie straina in relatia **PRODUS** (cod\_produș#, cod\_furnizor, luni\_garantie, pret, nume), inlocuind atributul pret. Aceasta structura ar permite ca produsele sa nu aibă doar un pret universal in cadrul lantului de magazine, putandu-se gasi la preturi diferite.

Din punct de vedere al memoriei utilizate, nu există o diferenta semnificativa intre structura de dinainte de denormalizare si dupa denormalizare. Desi o valoare a cheii din tabela STOC poate fi atribuita ca si cheie straina mai multor produse cu acelasi pret (retinand o singura data valoarea unui pret) tot este nevoie sa retinem o informatie despre pret in cadrul tabelii **PRODUS** (fie ca este valoarea pretului in sine sau o referinta catre tabela stoc).

Daca am aplica denormalizarea, ar insemna sa renunțăm la tabela stoc si sa ne intoarcem la structura initiala a diagramei, unde există un pret unic pentru fiecare produs si acesta este stocat chiar in tabela PRODUS. Aceasta operatie ar permite accesarea cu mai multa usurinta a pretului unui produs, nefiind necesar lucrul cu tabela suplimentara STOC.

In figura se poate observa structura inainte si dupa denormalizare.



Daca am vrea sa afisam pretul pentru fiecare produs, ar trebui sa scriem urmatoarea interogare:

*-inainte de denormalizare:*

```
select pret
from stoc s join produs p on (s.cod_stoc = p.cod_stoc);
```

*-dupa denormalizare:*

```
select pret
from produs;
```