PROIECT BAZE DE DATE

Cochior Iulia-Ștefana

Grupa 144

Cuprins

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare..............................3

2. Prezentarea constrângerilor impuse asupra modelului...............................................................3

3. Descrierea entităților, incluzând precizarea cheii primare.........................................................4

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora..............................................5

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri...............................6

6. Realizarea diagramei entitate-relație........................................................................................10

7. Realizarea diagramei conceptuale............................................................................................10

8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale...........................11

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).................................................11

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.........................14

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea...................15

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe.....................................28

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.......36

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

Baza de date conține informații referitoare la funcționarea unui buncăr, camerele și departamentele acestuia, persoanele ce locuiesc și muncesc în interior, cât și aprovizionarea și resursele consumate. Scopul creării acestei baze de date este de a putea ține evidența resurselor si a repartizarii pe departamente.

În tabelul Oameni, sunt înregistrate informații despre fiecare persoană, conținând următoarele date: nume, prenume, vârstă și satisfacția profesională. Fiecare persoana este repartizată într-o camera, care aparține unui departament. Departamentele au mai multe camere.

În tabelul Camere, sunt înregistrate detalii despre fiecare cameră din buncar. Acest tabel conține următoarele informații: numele camerei, capacitatea camerei exprimată în numărul maxim de persoane ce pot munci în aceasta și departamentul de care aparține.

În tabelul Departamente, sunt incluse următoarele informații: numele fiecărui departament și numele șefului de departament.

În tabelul Consumație, sunt înregistrate toate resursele consumate de fiecare departament, stocurile prezente, data consumarii și departamentul care a consumat resursa. Tabelul Aprovizionare conține informații despre cantitățile de resurse livrate și data aprovizionărilor.

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

O persoana lucrează într-o cameră iar într-o cameră pot lucra mai multe persoane.

O cameră aprține unui departament, iar într-un departament sunt mai multe camere.

O persoană este șef într-un singur departament, iar un departament are un șef.

Un departament poate avea mai multe consumații, iar o consumație aparține unui unic departament, deoarece nu ne focusăm pe resursa consumată, care poate să se repete, ci pe actul consumării, identificat printr-un id unic.

O consumație folosește o resursă, iar o resursă apare în mai multe consumații.

Aprovizionarea aduce o resursă, iar o resursă poate fii aprovizionată de mai multe ori.

3. Descrierea entităților, incluzând precizarea cheii primare.

Entitate: OAMENI

-informații generale despre oamenii din buncăr

-cheia primară este ID

Entitate: DEPARTAMENTE

-împărțirea buncărului după responsabilități ; fiecare departament are un nume și un șef și este compus din mai multe camere

-cheia primară este ID\_Departament

Entitate: CAMERE

-fiecare cameră are un nume, o capacitate maximă și un departament de care aparține -cheia primară este ID\_Cameră

Entitate: DISTRIBUȚIE\_CAMERE

-acest tabel are rolul de a atribui fiecărei persoane câte o cameră

-cheia primară este compusă din două chei străine: ID\_Cameră și ID\_Persoană

Entitate: RESURSE

-proviziile de care dispune buncărul și cantitățile în care acestea se gâsesc

-cheia primară este ID\_Resursă

Entitate: APROVIZIONARE

-ține evidența resurselor ce au fost primite, cantitatea primită și data aprovizionări

\_cheia primară este ID\_Aprovizionare

Entitate: CONSUMAȚIE

-ține evidența modului în care sunt utilizate resursele, cantitatea folosită, departamentul consumator precum și data la care are loc consumația

-cheia primară este ID\_Consumație

4.Descrierea relațiilor, incluzând precizarea cardinalității acestora.

OM\_este\_șef\_în\_DEPARTAMENT

-relație de tip one to one care leagă entitățile OAMENI și DEPARTAMENTE. Ea are cardinaliate minimă 1:1 (un om este șef într-un departament și fiecare departament are un șef) și cardinalitate maximă 1:1 (într-un departament există un șef, iar un om poate fii șef într-un singur departament).

OM\_are\_DISTRIBUȚIE\_CAMERĂ

-relație de tip one to one care leagă entitățile OAMENI și DISTRIBUȚIE\_CAMERE. Ea are cardinalitate minimă 1:1 (fiecărui om îi este atribuită o cameră și fiecare cameră are un om) și cardinalitate maximă 1:1 (o distribuție alocă un singur om, iar un om poate lucra doar într-o cameră).

CAMERĂ\_în\_DISTRIBUȚIE\_CAMERE

-relație de tip one to many care leagă entitățile CAMERĂ și DISTRIBUȚIE\_CAMERE. Ea are cardinalitatea minimă 1:1 ( o distribuție alocă o cameră, iar o cameră apare în cel puțin o distribuție) și maximă 1:n (o distribuție alocă doar o cameră, dar o cameră se gâsește în mai multe distribuții).

DEPARTAMENT\_format\_din\_CAMERE

-relație de tip one to many care leagă entitățile DEPARTAMENTE și CAMERE. Ea are cardinalitatea minimă 1:1 (un departament are cel puțin o cameră și o cameră aparține unui singur departament) și maximă 1:n (un departament poate avea mai multe camere, dar o cameră aparține unui singur departament)

DEPARTAMENT\_face\_CONSUMATIE

-relație de tip one to many care leagă entitățile DEPARTAMENTE și CONSUMAȚIE. Ea are cardinalitatea minimă 1:1 (un departament are minim o consumație și o consumație poate fii înregistrată într-un singur departament) și 1:n (un departament poate avea mai multe consumații dar o consumație este înregistrată într-un singur departament)

DEPARTAMENT\_face\_APROVIZIONARE

-relație de tip one to many care leagă entitățile DEPARTAMENTE și APROVIZIONARE. Ea are cardinalitatea minimă 1:1 (un departament are minim o aprovizionare și o aprovizionare poate fii înregistrată într-un singur departament) și 1:n (un departament poate avea mai multe aprovizionări dar o aprovizionare este înregistrată într-un singur departament)

CONSUMAȚIE\_folosește\_RESURSE

-relație de tip many to one care leagă entitățile CONSUMAȚIE și RESURSE. Ea are cardinalitatea minimă 1:1 (o consumație folosește minim o resursă și o resursă este folosită în minim o consumație) și maximă 1:n (o consumație are o singură resursă <<avem în tabelul Consumție FOREIGN KEY ID\_Resursă >> și o resursă poate fii folosită în mai multe consumații).

APROVIZIONARE\_aduce\_RESURSE

-relație de tip many to one care leagă entitățile APROVIZIONARE și RESURSE. Ea are cardinalitatea minimă 1:1 (o aprovizionare folosește minim o resursă și o resursă este folosită în minim o aprovizionare) și maximă 1:n (o aprovizionare aduce o singură resursă <<avem în tabelul Aprovizionare FOREIGN KEY ID\_Resursă >> și o resursă poate apărea în mai multe aprovizionări).

5.. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

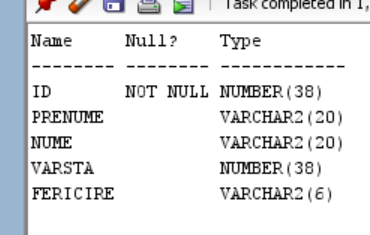
**Entitatea OAMENI are ca atribute:**

ID = variabilă de tip întreg, care reprezintă ID-ul unui om din buncăr

Nume și Prenume = variabile de tip caracter, de lungime maximă 20, care reprezintă numele unui om

Vârstă = variabilă de tip întreg, care reprezintă vârsta unui om

Fericire = variabilă de tip caracter, de lungime maximă 6, care reprezintă nivelul de fericire al unui om, exprimat în procente (ex. “78%” )

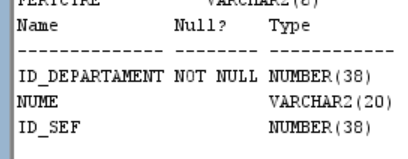


**Entitatea DEPARTAMENTE are ca atribute:**

Departament\_ID = variabilă de tip întreg, care reprezintă ID-ul unui departament din buncăr

Nume = variabilă de tip caracter, de lungime maximă 20, care reprezintă denumirea departamentului

ID\_Sef = variabilă de tip întreg, care reprezintă ID-ul responsabilului de departament; corespunde unei valori a cheii primare din tabelul OAMENI.



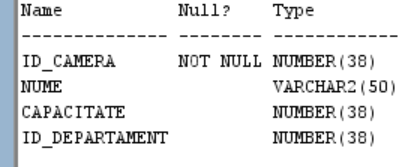
**Entitatea CAMERE are ca atribute:**

ID\_Cameră = variabilă de tip întreg, care reprezintă ID-ul unei camere dintr-un departament

Nume = variabilă de tip caracter, de lungime maximă 20, care reprezintă denumirea camerei

Capacitate = variabilă de tip întreg, care reprezintă capacitatea maximă (de oameni) ce pot lucra într-o cameră

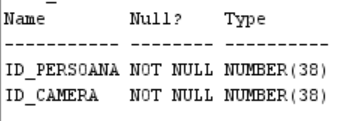
ID\_Departament = variabilă de tip întreg, care reprezintă ID-ul departamentului în care se află camera; corespunde unei valori a cheii primare din tabelul DEPARTAMENTE



**Entitatea DISTRIBUȚIE\_CAMERE are ca atribute:**

ID\_Cameră = variabilă de tip întreg, care reprezintă ID-ul camerei în care a fost repartizată persoana; corespunde unei valori a cheii primare din tabelul CAMERE

ID\_Persoană = variabilă de tip întreg, care reprezintă ID-ul persoanei care a fost repartizată; corespunde unei valori a cheii primare din tabelul OAMENI



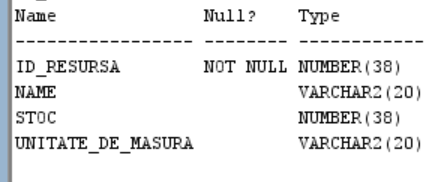
**Entitatea RESURSE are ca atribute:**

ID\_Resursă = variabilă de tip întreg, care reprezintă ID-ul resursei

Nume = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele resursei

Stoc = variabilă de tip întreg, care reprezintă cantitatea de resursă aflată în stoc

Unitate\_de\_măsură = variabilă de tip caractre, de lungime maximă 20, care reprezintă în ce se măsoară cantitatea de resursă (ex: bidoane, pachete, ...)



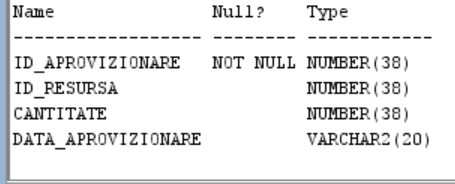
**Entitatea APROVIZIONARE are ca atribute:**

ID\_Aprovizionare = variabilă de tip întreg, care reprezintă ID-ul aprovizionării

ID\_Resursă = variabilă de tip întreg, care reprezintă ID-ul resursei care a fost aprovizionată; corespunde unei valori a cheii primare din tabelul RESURSE

Cantitate = variabilă de tip întreg, care reprezintă cantitatea de resursă primită

Data\_Aprovizionare = variabilă de tip caracter, de lungime maxima 20, care reprezintă data la care a avut loc aprovizionarea



**Entitatea CONSUMAȚIE are ca atribute:**

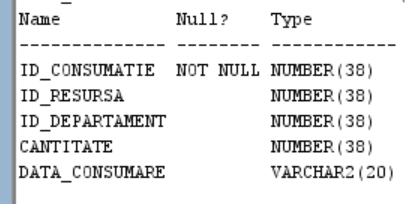
ID\_Consumație= variabilă de tip întreg, care reprezintă ID-ul consumației

ID\_Resursă = variabilă de tip întreg, care reprezintă ID-ul resursei care a fost consumată; corespunde unei valori a cheii primare din tabelul RESURSE

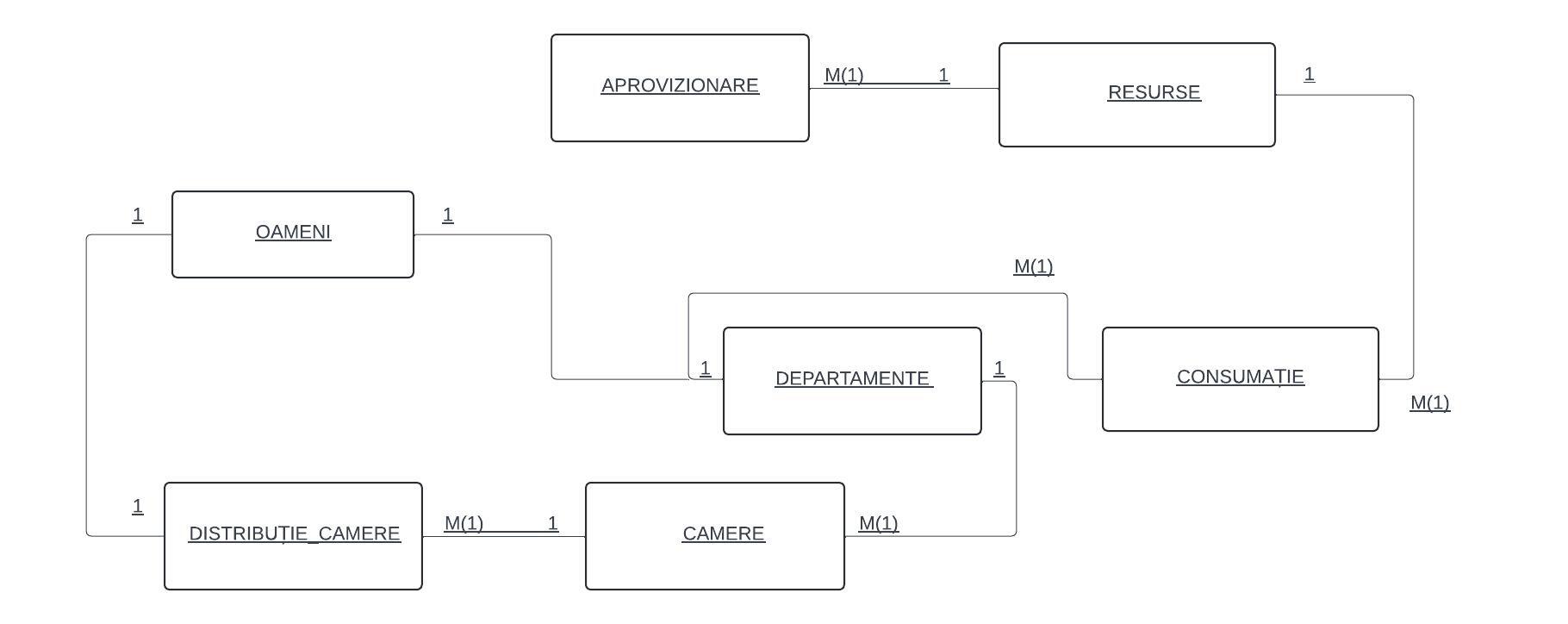
ID\_Departament = valoare de tip întreg, care reprezintă ID-ul departamentului care a consumat resursa; corespunde unei valori a cheii primare din tabelul DEPARTAMENTE

Cantitate = variabilă de tip întreg, care reprezintă cantitatea de resursă consumată

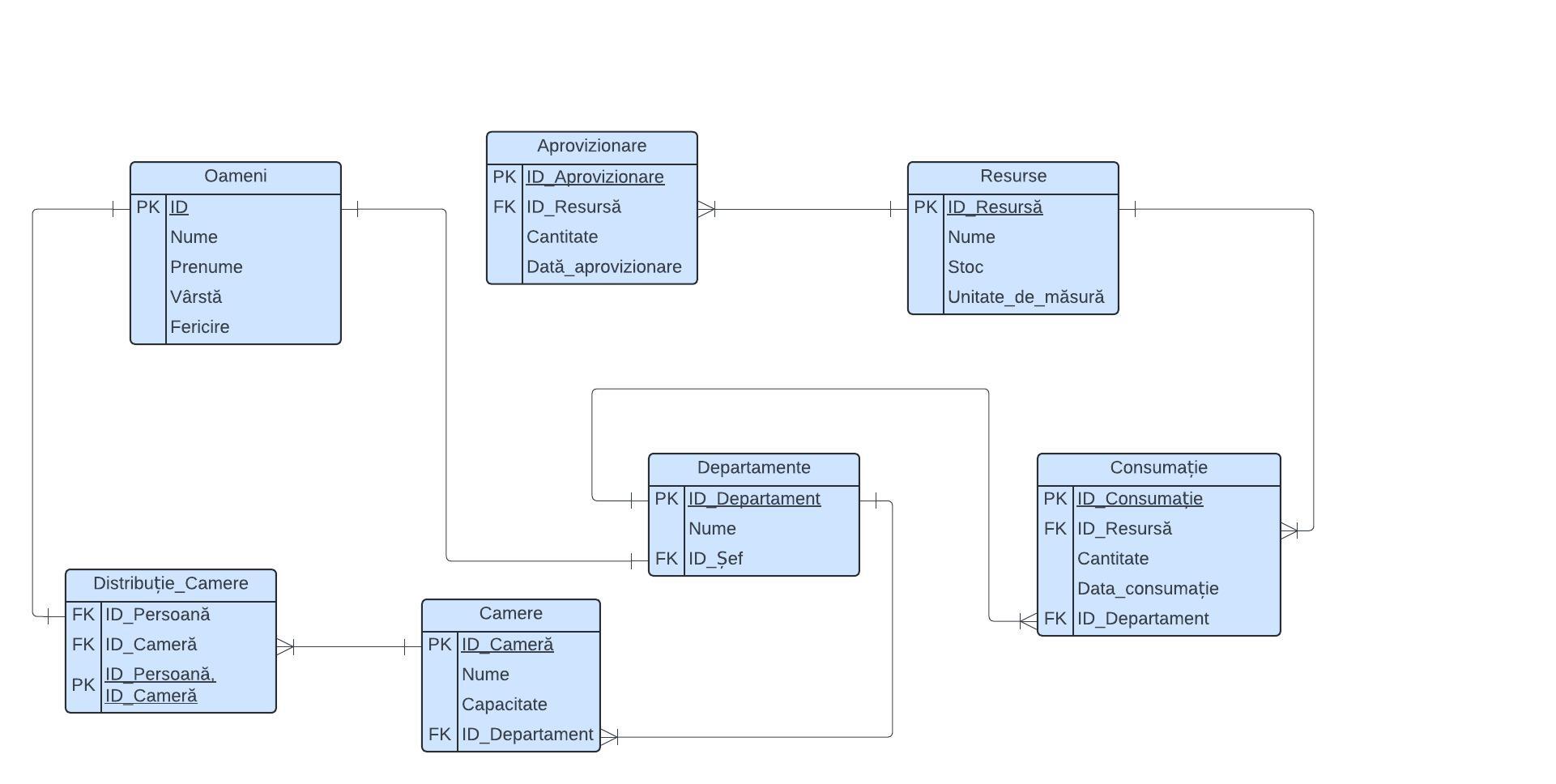
Data\_Consumare = variabilă de tip caracter, de lungime maxima 20, care reprezintă data la care a avut loc consumarea



6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Schemele relaţionale corespunzătoare diagramei conceptuale sunt:

OAMENI (ID#, Nume, Prenume, Vârstă, Fericire)

DEPARTAMENTE (ID\_Departament#, Nume, ID\_Sef)

CAMERE (ID\_Cameră#, Nume, Capacitate, ID\_Departament)

DISTRIBUȚIE\_CAMERE (ID\_Persoană#, ID\_Cameră#)

RESURSE (ID\_Resursă#, Nume, Stoc, Unitate\_de\_măsură)

APROVIZIONARE (ID\_Aprovizionare#, ID\_Resursă, Cantitate, Dată\_Aprovizionare)

CONSUMAȚIE (ID\_Consumație#, ID\_Resursă, ID\_Departament, Cantitate, Dată\_Consumație)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

**Forma normală 1 (FN1)**

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizabilă. Forma normală 1 impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare. În cadrul bazei de date construite de mine voi ilustra un exemplu cu entitațile CONSUMAȚIE și DEPARTAMENT.

Pentru o consumație sunt listate toate departamentele care au consumat aceeași resursă:

|  |  |
| --- | --- |
| ID\_Resursă | ID\_Departament |
| 726 | 1, 2, 4 |
| 732 | 2, 3 |

A selecta interegistrări pe baza câmpurilor care pot conține valori semnificative este foarte dificil si greu de implementat. Spre exemplu, o interogare care ar selecta resursele care au fost consumate de departamentul cu ID 2, ar trebui să parcurgem fiecare șir ”ID\_Departament”, să identificăm subșirul 2 și să selectăm numai acele înregistrări în care apare acest subșir. (ID-urile ar fii tinute sub formă de șir de caractere).

Tabelul normalizat arată în felul următor:

|  |  |  |
| --- | --- | --- |
| ID\_Consumație | ID\_Resursă | ID\_Departament |
| 1 | 726 | 1 |
| 2 | 726 | 2 |
| 3 | 726 | 4 |
| 4 | 732 | 2 |
| 5 | 732 | 3 |

**Forma normală 2 (FN2)**

O relație se află în FN2 dacă și numai dacă această relație este deja în FN1, iar fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară. Știm că FN2 presupune să nu existe dependențe funcționale parțiale în cadrul relației.

Voi prezenta un exemplu pentru tabelul DISTRIBUIRE\_CAMERE.

|  |  |  |
| --- | --- | --- |
| **ID\_Persoană** | **ID\_Cameră** | **Nume\_Cameră** |
| **1022** | **102** | **Camera de filtrare** |
| **1028** | **601** | **Distribuire** |

Relația este în NF1 deoarece avem identificator unic pentru toate intrările din tabel.

Fiecare atribut care nu este cheie primară (cheia primară este compusă: ID\_Persoană și ID\_Cameră) este dependent de întreaga cheie primară, dar în cazul nostru atributul Nume\_Cameră nu depinde direct de întreaga cheie primară deoarece se observă dependența directă dintre Nume\_Cameră și ID\_Persoană însemnând că Nume\_Cameră depinde direct doar de o parte a cheii primare, si anume doar de ID\_Persoană, de unde se deduce că relația nu se află în FN2.

Astfel, avem următorul tabel normalizat:

|  |  |
| --- | --- |
| ID\_Persoană | ID\_Cameră |
| 1022 | 102 |
| 1028 | 601 |

**Forma normală 3 (FN3)**

O relație se află în FN3 dacă și numai dacă această relație este deja in FN2 (și implicit în FN1), iar fiecare atribut care nu este cheie primară este dependent de cheia primară.

O posibilă formă inițială a tabelelor OAMENI și CAMERE este:

OAMENI (tabelele sunt simplificate pentru exemplu)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nume | Prenume | ID\_Departament |
| 1 | Nume1 | Prenume1 | 1 |
| 2 | Nume2 | Prenume2 | 2 |
| 3 | Nume3 | Prenume3 | 1 |

**CAMERE**

|  |  |  |
| --- | --- | --- |
| ID\_Cameră | Nume | ID\_Departament |
| 101 | Camera 1 | 1 |
| 102 | Camera 2 | 2 |

Problema aici este că ID\_Departament este o dependență tranzitivă. În acest caz ID\_Departament depinde de ID\_Cameră, care la rândul ei depinde de ID\_Departament (o cameră poate aparține unui unic departament). Aceasta este o formă de dependență tranzitivă care nu respectă forma normală 3. Avem nevoie de un mod de a distribui camerele către oameni. Vom crea tabelul Distribuție\_Camere pentru a gestiona relația dintre camere și oameni. Tabele în forma finală FN3 arată în felul următor:

OAMENI

|  |  |  |
| --- | --- | --- |
| ID | Nume | Prenume |
| 1 | Nume1 | Prenume1 |
| 2 | Nume2 | Prenume2 |
| 3 | Nume3 | Prenume3 |

CAMERE

|  |  |
| --- | --- |
| ID\_Cameră | Nume |
| 1 | Camera 1 |
| 2 | Camera 2 |

DISTRIBUȚIE\_CAMERE

|  |  |
| --- | --- |
| ID\_Persoană | ID\_Cameră |
| 1 | 101 |
| 2 | 102 |
| 3 | 101 |

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

Cod SQL:

CREATE SEQUENCE ID\_SEQ START WITH 1022;

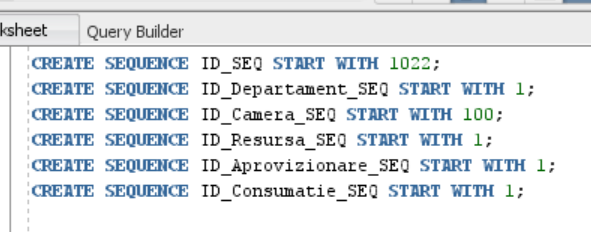
CREATE SEQUENCE ID\_Departament\_SEQ START WITH 1;

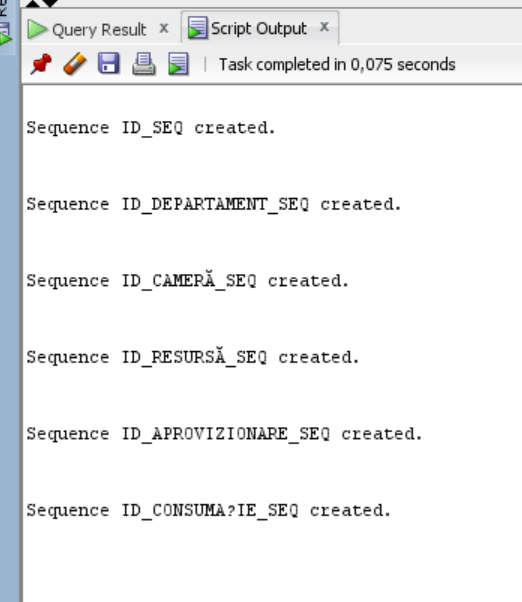
CREATE SEQUENCE ID\_Cameră\_SEQ START WITH 100;

CREATE SEQUENCE ID\_Resursă\_SEQ START WITH 1;

CREATE SEQUENCE ID\_Aprovizionare\_SEQ START WITH 1;

CREATE SEQUENCE ID\_Consumație\_SEQ START WITH 1;





11.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

CREATE TABLE Oameni(

ID INT DEFAULT ID\_SEQ.NEXTVAL PRIMARY KEY,

Prenume VARCHAR(20),

Nume VARCHAR(20),

Varsta INT,

Fericire VARCHAR(6)

);

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Rachel', 'Day', 24, '75%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Randy', 'Walker', 47, '77%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Ashley', 'Graham', 33, '67%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Teresa', 'Martin', 29, '69%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES( 'Jack', 'Sutton', 43, '73%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Angela', 'West', 40, '70%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Daniel', 'Saubers', 25, '90%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Alan', 'Smith', 27, '85%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Ethan', 'Browm', 30, '80%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Ava', 'Davis', 26, '91%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Emma', 'Brown', 22, '87%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Liam', 'Anderson', 35, '80%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Lottie', 'Caprice', 12, '98%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Marian', 'Martescu', 37, '81%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Lester', 'Anderson', 56, '77%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Holly', 'Jefferson', 36, '89%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Clementine', 'Barny', 24, '78%');

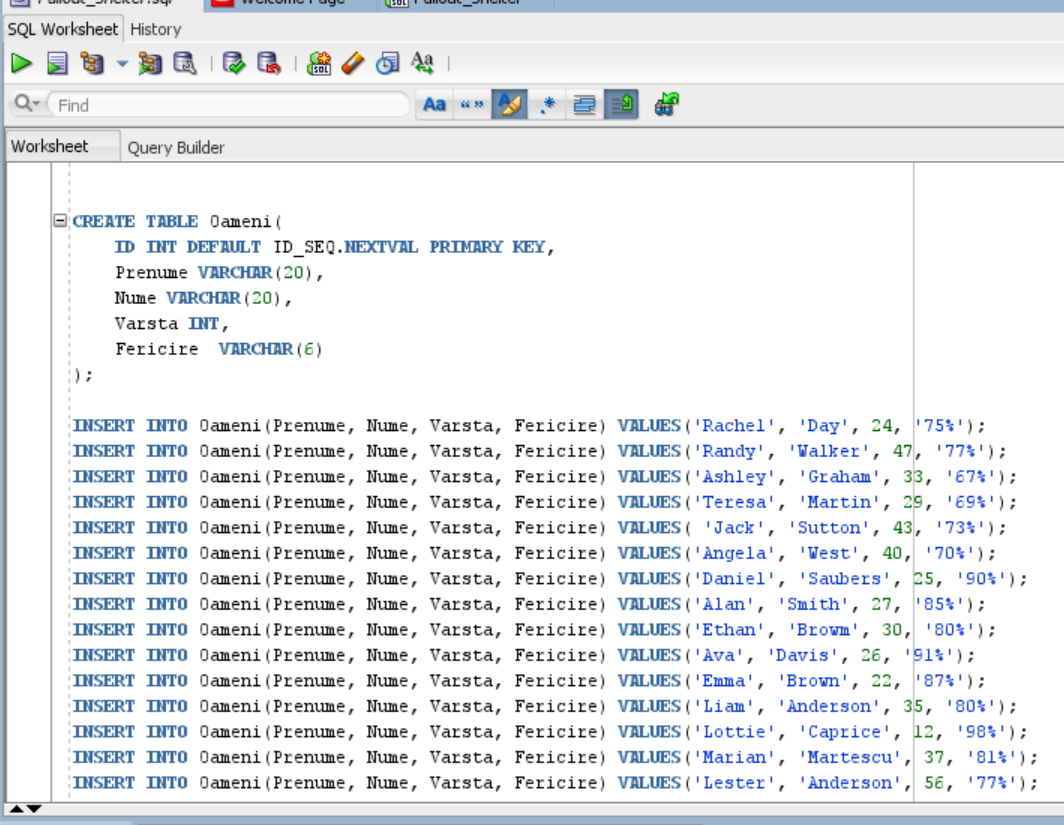
INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Pamela', 'Albine', 28, '87%');

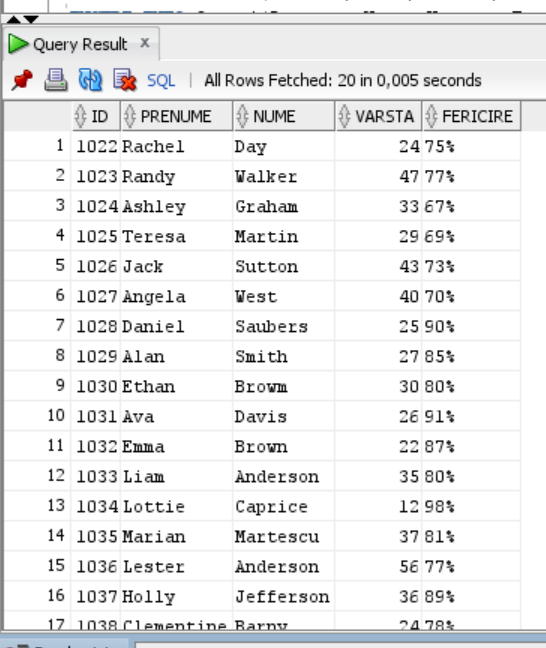
INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Xenon', 'Lee Park', 31, '85%');

INSERT INTO Oameni(Prenume, Nume, Varsta, Fericire) VALUES('Darian', 'Constance', 33, '70%');

SELECT \*

FROM Oameni;





CREATE TABLE Departamente(

ID\_Departament INT DEFAULT ID\_Departament\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR(20),

ID\_Sef INT,

FOREIGN Key(ID\_Sef) REFERENCES Oameni(ID)

);

INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Tratarea Apei', 1023);

INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Generare Electrica', 1025);

INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Cantina', 1026);

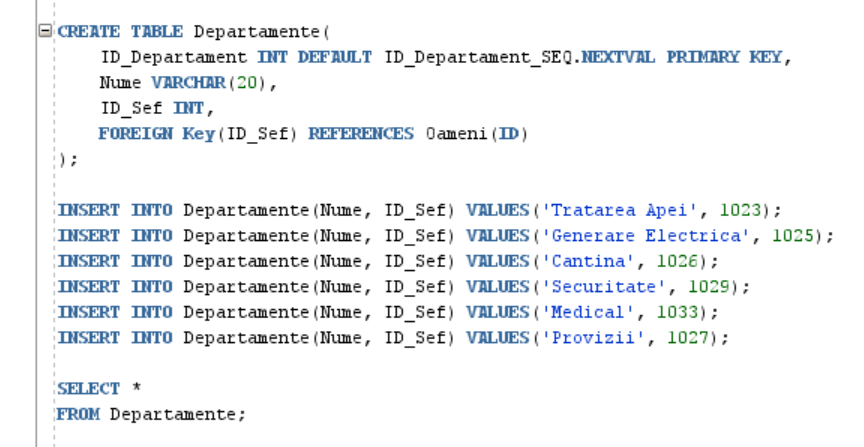
INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Securitate', 1029);

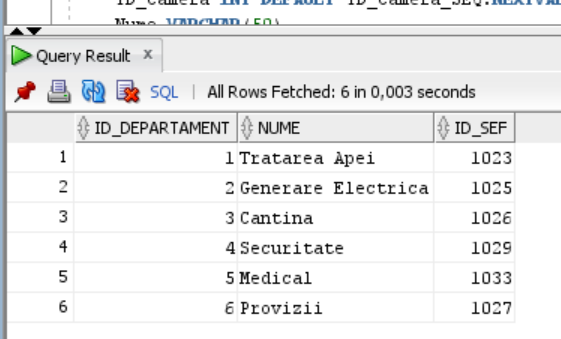
INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Medical', 1033);

INSERT INTO Departamente(Nume, ID\_Sef) VALUES('Provizii', 1027);

SELECT \*

FROM Departamente;





CREATE TABLE Camere (

ID\_Camera INT DEFAULT ID\_Camera\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR(50),

Capacitate INT,

ID\_Departament INT,

FOREIGN KEY (ID\_Departament) REFERENCES Departamente(ID\_Departament)

);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Controlul Calitatii', 3, 1);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Centru de Distribuire', 10, 1);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Camera de Filtrare', 4, 1);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Depozitare Combustibil', 3, 2);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Panou de Electricitate', 10, 2);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Generator de Urgenta', 5, 2);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Bucatarie', 10, 3);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Sala de Mese', 4, 3);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Depozit de Mancare', 5, 3);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Supraveghere Video', 3, 4);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Control de securitate', 5, 4);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Farmacie', 2, 5);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Camera de Examinare', 2, 5);

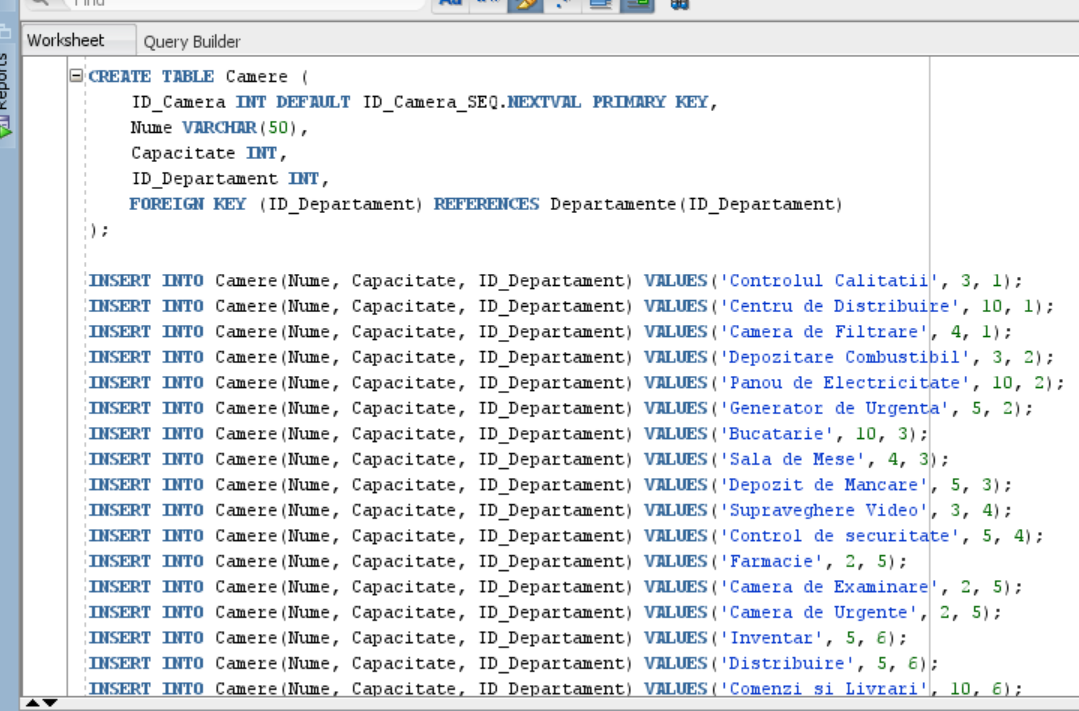
INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Camera de Urgente', 2, 5);

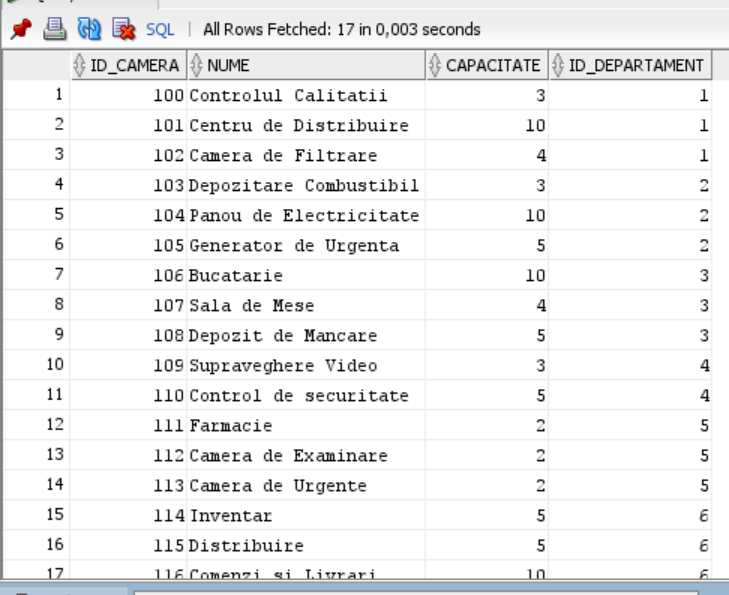
INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Inventar', 5, 6);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Distribuire', 5, 6);

INSERT INTO Camere(Nume, Capacitate, ID\_Departament) VALUES('Comenzi si Livrari', 10, 6);

SELECT \* FROM Camere;





CREATE TABLE Resurse(

ID\_Resursa INT DEFAULT ID\_Resursa\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR(20),

Stoc INT,

Unitate\_de\_masura VARCHAR(20)

);

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Conserve', 200, 'cutii');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Apa', 500, 'bidoane');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Trusa de prim ajutor', 10, 'truse');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Medicamente', 50, 'sticle');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Saci de dormit', 30, 'bucati ');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Extinctoare', 5, 'bucati');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Masca de gaze', 15, 'masti');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Baterii', 200, 'pachete');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Combustibil', 500, 'canistre');

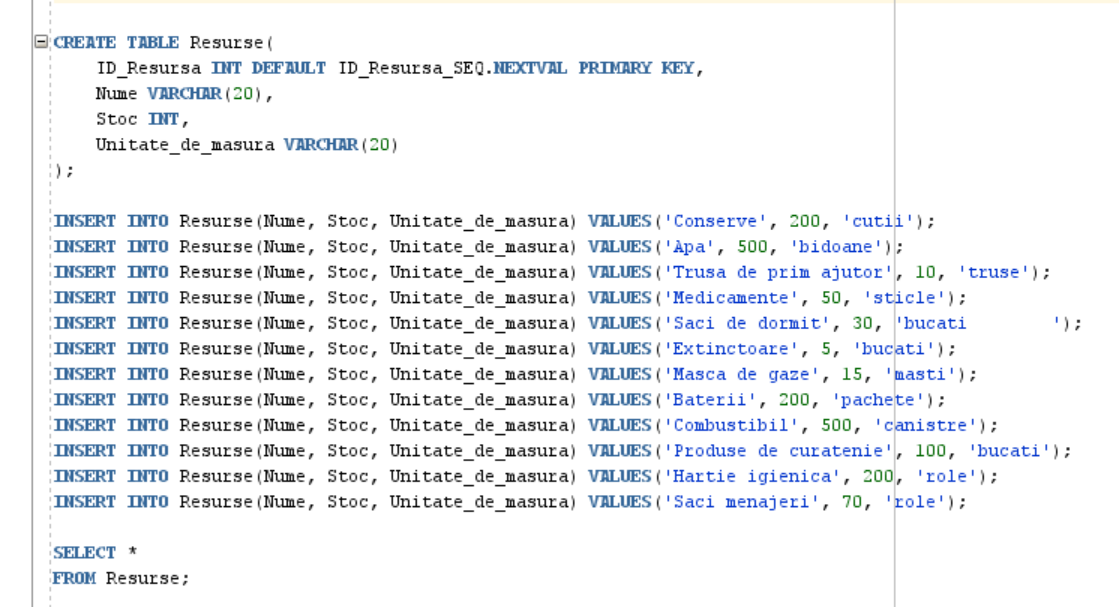
INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Produse de curatenie', 100, 'bucati');

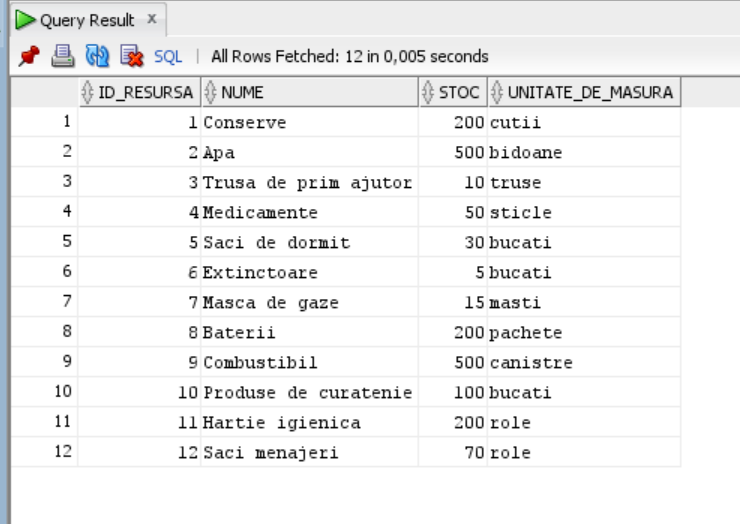
INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Hartie igienica', 200, 'role');

INSERT INTO Resurse(Nume, Stoc, Unitate\_de\_masura) VALUES('Saci menajeri', 70, 'role');

SELECT \*

FROM Resurse;





CREATE TABLE Distributie\_Camere (

ID\_Persoana INT,

ID\_Camera INT,

PRIMARY KEY (ID\_Persoana, ID\_Camera),

FOREIGN KEY (ID\_Persoana) REFERENCES Oameni(ID),

FOREIGN KEY (ID\_Camera) REFERENCES Camere(ID\_Camera)

);

INSERT INTO Distributie\_Camere VALUES(1029, 100);

INSERT INTO Distributie\_Camere VALUES(1030, 101);

INSERT INTO Distributie\_Camere VALUES(1025, 102);

INSERT INTO Distributie\_Camere VALUES(1023, 103);

INSERT INTO Distributie\_Camere VALUES(1026, 104);

INSERT INTO Distributie\_Camere VALUES(1028, 105);

INSERT INTO Distributie\_Camere VALUES(1032, 106);

INSERT INTO Distributie\_Camere VALUES(1027, 107);

INSERT INTO Distributie\_Camere VALUES(1033, 108);

INSERT INTO Distributie\_Camere VALUES(1022, 109);

INSERT INTO Distributie\_Camere VALUES(1024, 110);

INSERT INTO Distributie\_Camere VALUES(1031, 112);

INSERT INTO Distributie\_Camere VALUES(1034, 113);

INSERT INTO Distributie\_Camere VALUES(1035, 114);

INSERT INTO Distributie\_Camere VALUES(1036, 115);

INSERT INTO Distributie\_Camere VALUES(1037, 116);

INSERT INTO Distributie\_Camere VALUES(1038, 117);

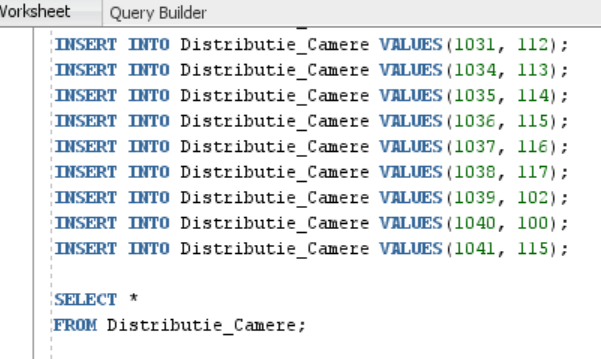
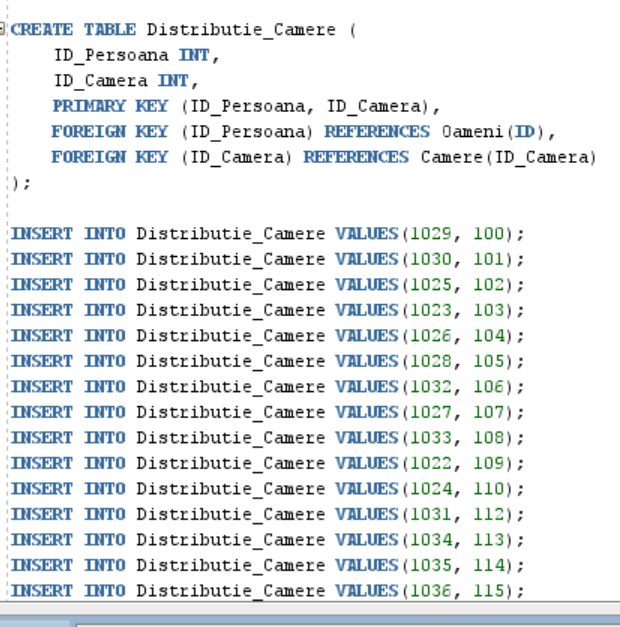
INSERT INTO Distributie\_Camere VALUES(1039, 102);

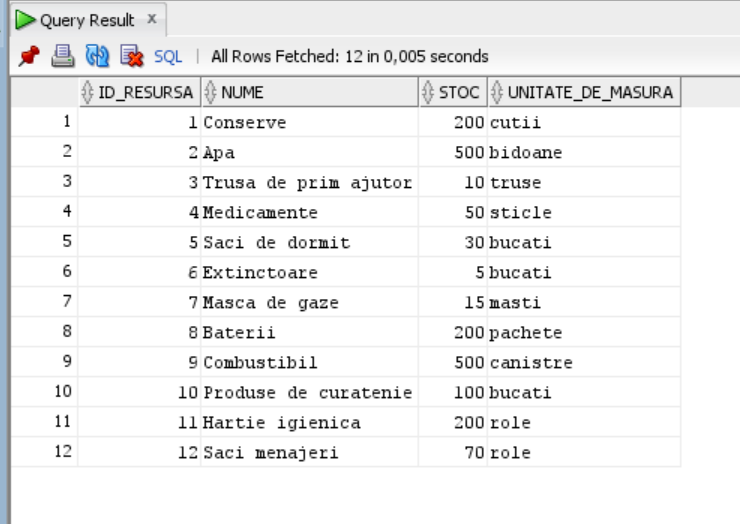
INSERT INTO Distributie\_Camere VALUES(1040, 100);

INSERT INTO Distributie\_Camere VALUES(1041, 115);

SELECT \*

FROM Distributie\_Camere;





CREATE TABLE Aprovizionare(

ID\_Aprovizionare INT DEFAULT ID\_Aprovizionare\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Resursa INT,

Cantitate INT,

Data\_aprovizionare VARCHAR(20),

FOREIGN KEY(ID\_Resursa) REFERENCES Resurse(ID\_Resursa)

);

INSERT INTO Aprovizionare(ID\_Resursa, Cantitate, Data\_Aprovizionare) VALUES(1, 100, '20.08.2089');

INSERT INTO Aprovizionare(ID\_Resursa, Cantitate, Data\_Aprovizionare) VALUES(7, 30, '01.08.2089');

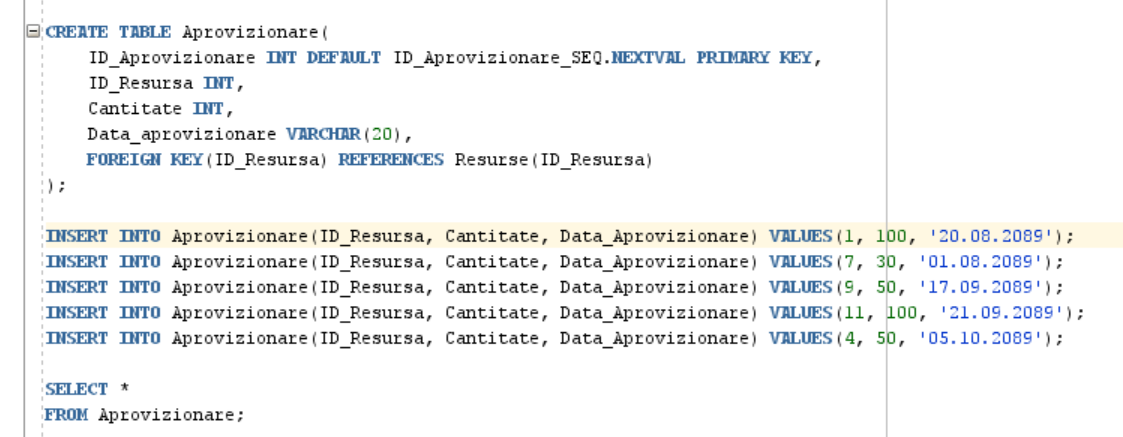
INSERT INTO Aprovizionare(ID\_Resursa, Cantitate, Data\_Aprovizionare) VALUES(9, 50, '17.09.2089');

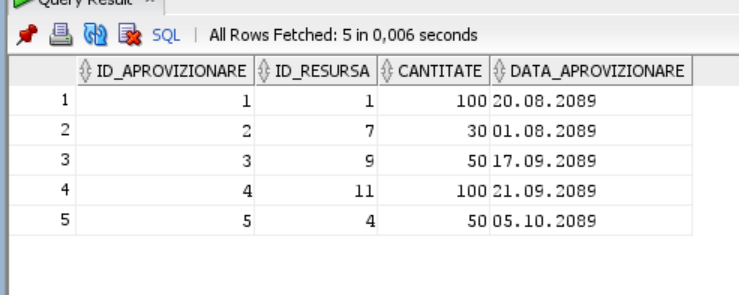
INSERT INTO Aprovizionare(ID\_Resursa, Cantitate, Data\_Aprovizionare) VALUES(11, 100, '21.09.2089');

INSERT INTO Aprovizionare(ID\_Resursa, Cantitate, Data\_Aprovizionare) VALUES(4, 50, '05.10.2089');

SELECT \*

FROM Aprovizionare;





CREATE TABLE Consumatie(

ID\_Consumatie INT DEFAULT ID\_Consumatie\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Resursa INT,

Cantitate INT,

Data\_consumatie VARCHAR(20),

ID\_Departament INT,

FOREIGN KEY(ID\_Resursa) REFERENCES Resurse(ID\_Resursa),

FOREIGN KEY(ID\_Departament) REFERENCES Departamente(ID\_Departament)

);

INSERT INTO Consumatie(ID\_Resursa, Cantitate, Data\_Consumatie, ID\_Departament) VALUES(2, 50, '06.10.2089', 2);

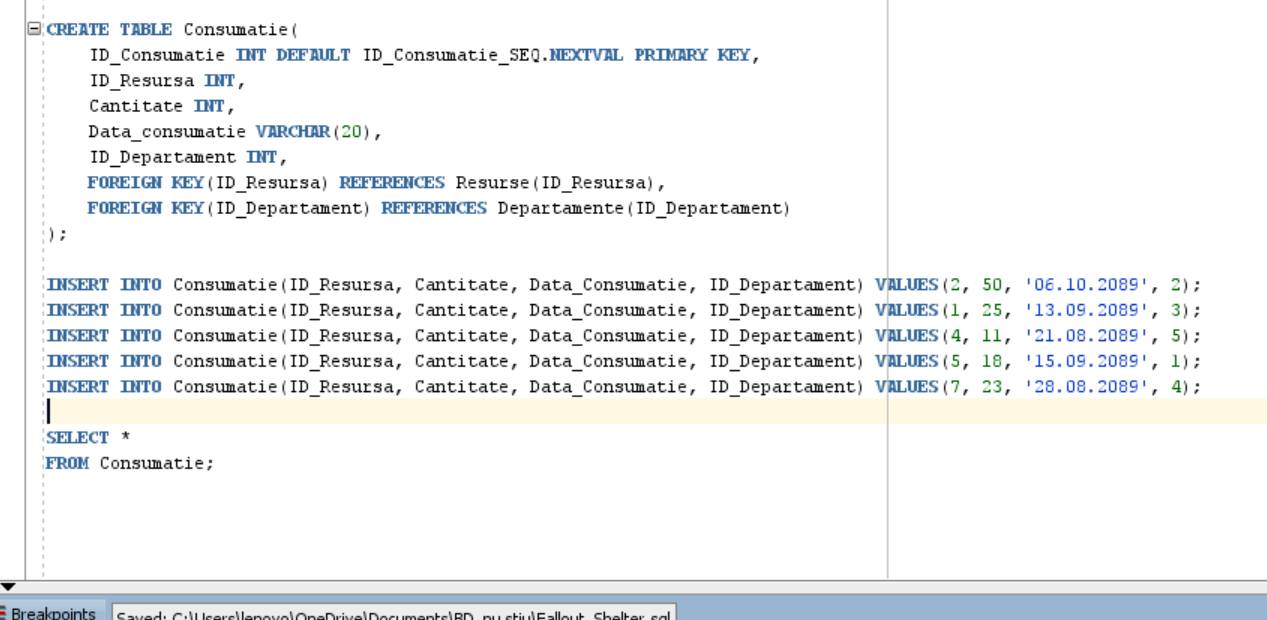
INSERT INTO Consumatie(ID\_Resursa, Cantitate, Data\_Consumatie, ID\_Departament) VALUES(1, 25, '13.09.2089', 3);

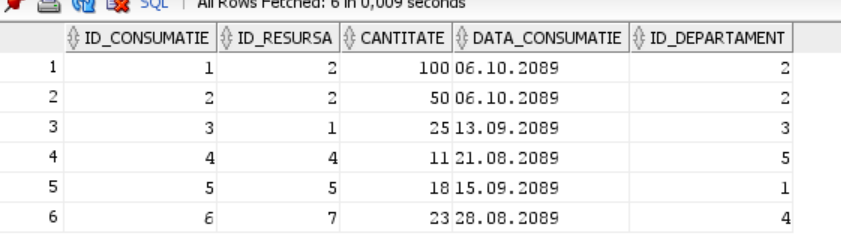
INSERT INTO Consumatie(ID\_Resursa, Cantitate, Data\_Consumatie, ID\_Departament) VALUES(4, 11, '21.08.2089', 5);

INSERT INTO Consumatie(ID\_Resursa, Cantitate, Data\_Consumatie, ID\_Departament) VALUES(5, 18, '15.09.2089', 1);

INSERT INTO Consumatie(ID\_Resursa, Cantitate, Data\_Consumatie, ID\_Departament) VALUES(7, 23, '28.08.2089', 4);

SELECT \* FROM Consumatie;





12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

a) subcereri sincronizate în care intervin cel puțin 3 tabele

b) subcereri nesincronizate în clauza FROM

c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**Cererea 1** (acoperă subpunctul a)

**Limbaj natural:** Găsiți toate resursele care au fost aprovizionate și consumate de cel puțin două ori, și au o cantitate totală mai mare de 100. Se va afișa id – ul resursei, numele și cantitatea.

SELECT ID\_Resursa, r.Nume, r.Stoc

FROM Resurse r

WHERE r.ID\_Resursa IN (

SELECT a.ID\_Resursa

FROM Aprovizionare a

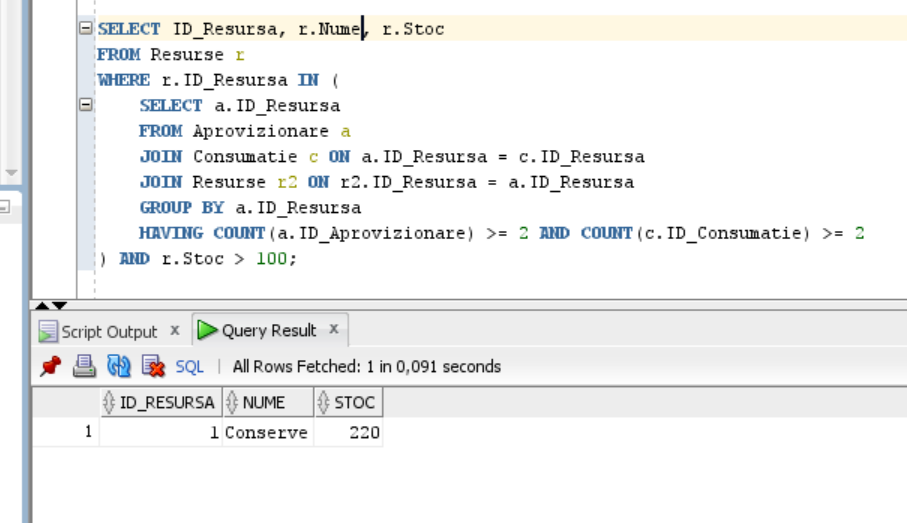
JOIN Consumatie c ON a.ID\_Resursa = c.ID\_Resursa

JOIN Resurse r2 ON r2.ID\_Resursa = a.ID\_Resursa

GROUP BY a.ID\_Resursa

HAVING COUNT(a.ID\_Aprovizionare) >= 2 AND COUNT(c.ID\_Consumatie) >= 2

) AND r.Stoc > 100;



**Cererea 2** (acoperă subpunctul b)

**Limbaj natural:** Găsiți resursele care au fost aprovizionate de cel puțin trei ori și consumațiile lor asociate. Se vor afișa numele resursei, stocul disponibil precum și consumațiile totale.

SELECT r.Nume, r.Stoc, NVL(sub.ConsumuriTotale, 0) AS ConsumuriTotale

FROM Resurse r

JOIN (

SELECT ID\_Resursa, SUM(Cantitate) AS ConsumuriTotale

FROM Consumatie

GROUP BY ID\_Resursa

) sub ON r.ID\_Resursa = sub.ID\_Resursa

WHERE r.ID\_Resursa IN (

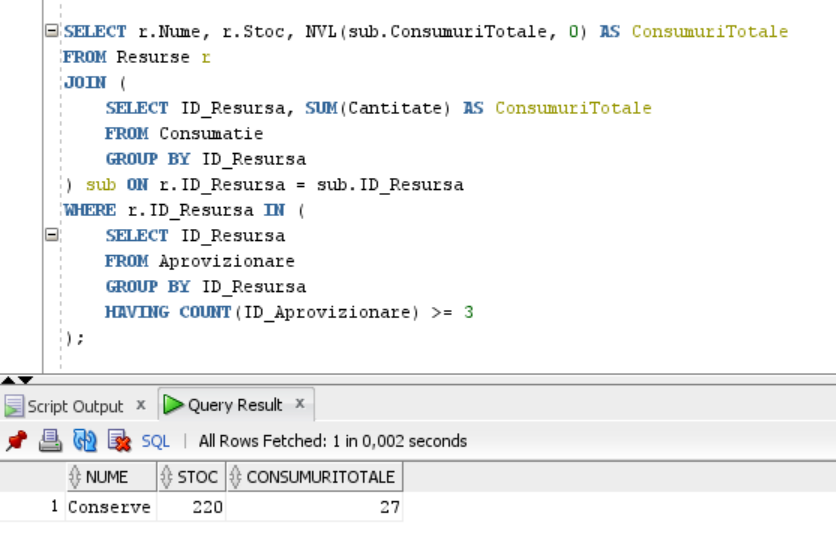
SELECT ID\_Resursa

FROM Aprovizionare

GROUP BY ID\_Resursa

HAVING COUNT(ID\_Aprovizionare) >= 3

);



**Cererea 3** (acoperă subpunctul c)

**Limbaj natural:** Găsiți departamentele care au aprovizionat resurse în anul 2089 și totalul resurselor consumate este mai mare de 100, precizând și numărul de resurse.

SELECT d.Nume, COUNT(DISTINCT c.ID\_Resursa) AS nr\_Resurse, SUM(c.Cantitate) AS TotalConsum

FROM Departamente d

JOIN Consumatie c ON d.ID\_Departament = c.ID\_Departament

JOIN Resurse r ON c.ID\_Resursa = r.ID\_Resursa

WHERE c.ID\_Resursa IN (

SELECT ID\_Resursa

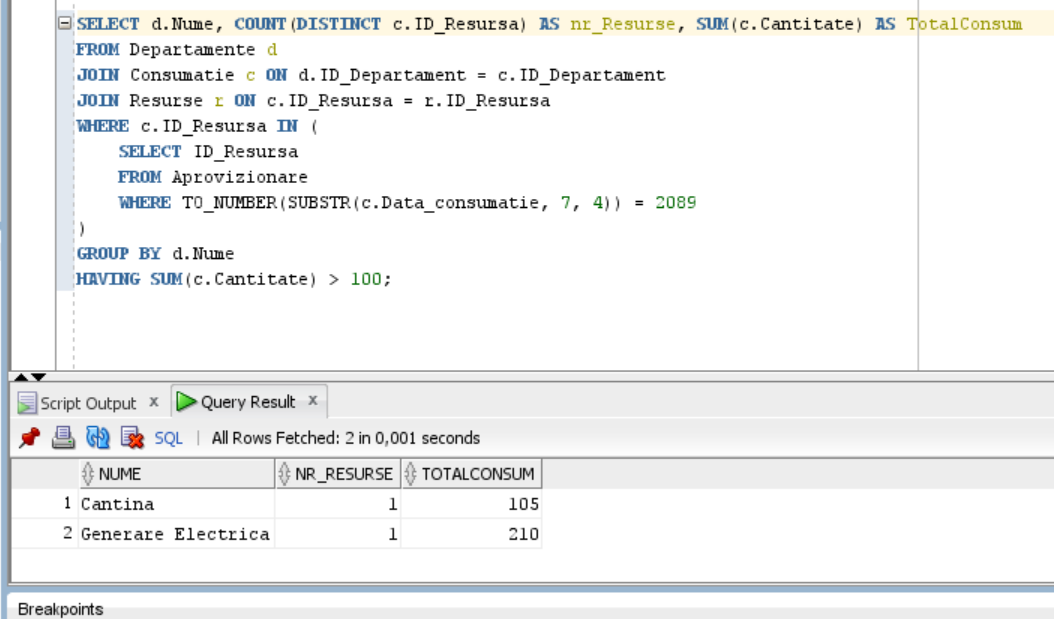
FROM Aprovizionare

WHERE TO\_NUMBER(SUBSTR(c.Data\_consumatie, 7, 4)) = 2089

)

GROUP BY d.Nume

HAVING SUM(c.Cantitate) > 100;



**Cererea 4** (acoperă subpunctul d)

**Limbaj natural:** Listați resursele și cantitatea lor, înlocuind cantitățile nule cu 0 și clasificând resursele în funcție de cantitate ("Scăzut", "Mediu", "Ridicat").

**SELECT r.Nume,**

**NVL(r.Stoc, 0) AS Cantitate,**

**DECODE(**

**CASE**

**WHEN NVL(r.Stoc, 0) <= 10 THEN 'Scăzut'**

**WHEN NVL(r.Stoc, 0) <= 50 THEN 'Mediu'**

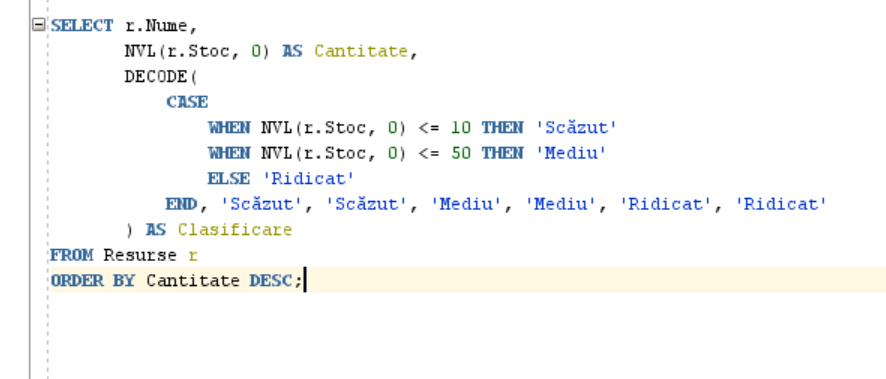
**ELSE 'Ridicat'**

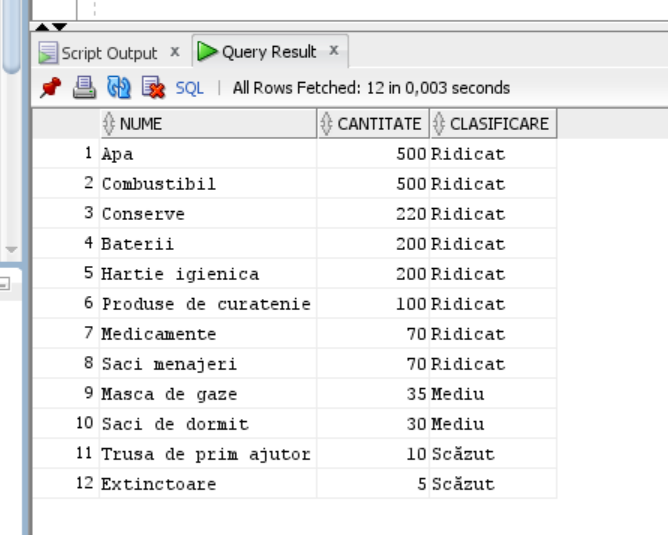
**END, 'Scăzut', 'Scăzut', 'Mediu', 'Mediu', 'Ridicat', 'Ridicat'**

**) AS Clasificare**

**FROM Resurse r**

**ORDER BY Cantitate DESC;**





**Cererea 5** (acoperă subpunctele e și f)

**Limbaj natural: S**electați numele fiecărei resurse, un prefix format din primele trei caractere ale numelui, numele resursei în majuscule, lungimea numelui resursei, data ultimei aprovizionări formatată în formatul 'DD/MM/YYYY', și clasificarea cantității totale consumate pentru fiecare resursă.

WITH RecentAprovizionare AS (

SELECT ID\_Resursa, MAX(Data\_Aprovizionare) AS Ultima\_Aprovizionare

FROM Aprovizionare

GROUP BY ID\_Resursa

),

TotalConsum AS (

SELECT c.ID\_Resursa, SUM(c.Cantitate) AS Total\_Cantitate

FROM Consumatie c

JOIN RecentAprovizionare ra ON c.ID\_Resursa = ra.ID\_Resursa

JOIN Aprovizionare a ON a.ID\_Resursa = c.ID\_Resursa AND a.Data\_Aprovizionare = ra.Ultima\_Aprovizionare

WHERE TO\_DATE(a.Data\_Aprovizionare, 'DD.MM.YYYY') >= TO\_DATE('01.01.2080', 'DD.MM.YYYY')

GROUP BY c.ID\_Resursa

)

SELECT

r.Nume,

SUBSTR(r.Nume, 1, 3) AS Prefix\_Nume,

UPPER(r.Nume) AS Nume\_Majuscule,

LENGTH(r.Nume) AS Lungime\_Nume,

TO\_CHAR(TO\_DATE(a.Data\_Aprovizionare, 'DD.MM.YYYY'), 'DD/MM/YYYY') AS Data\_Formatata,

CASE

WHEN t.Total\_Cantitate IS NULL THEN '0'

WHEN t.Total\_Cantitate <= 10 THEN 'Scăzut'

WHEN t.Total\_Cantitate <= 50 THEN 'Mediu'

ELSE 'Ridicat'

END AS Clasificare\_Cantitate

FROM

Resurse r

LEFT JOIN

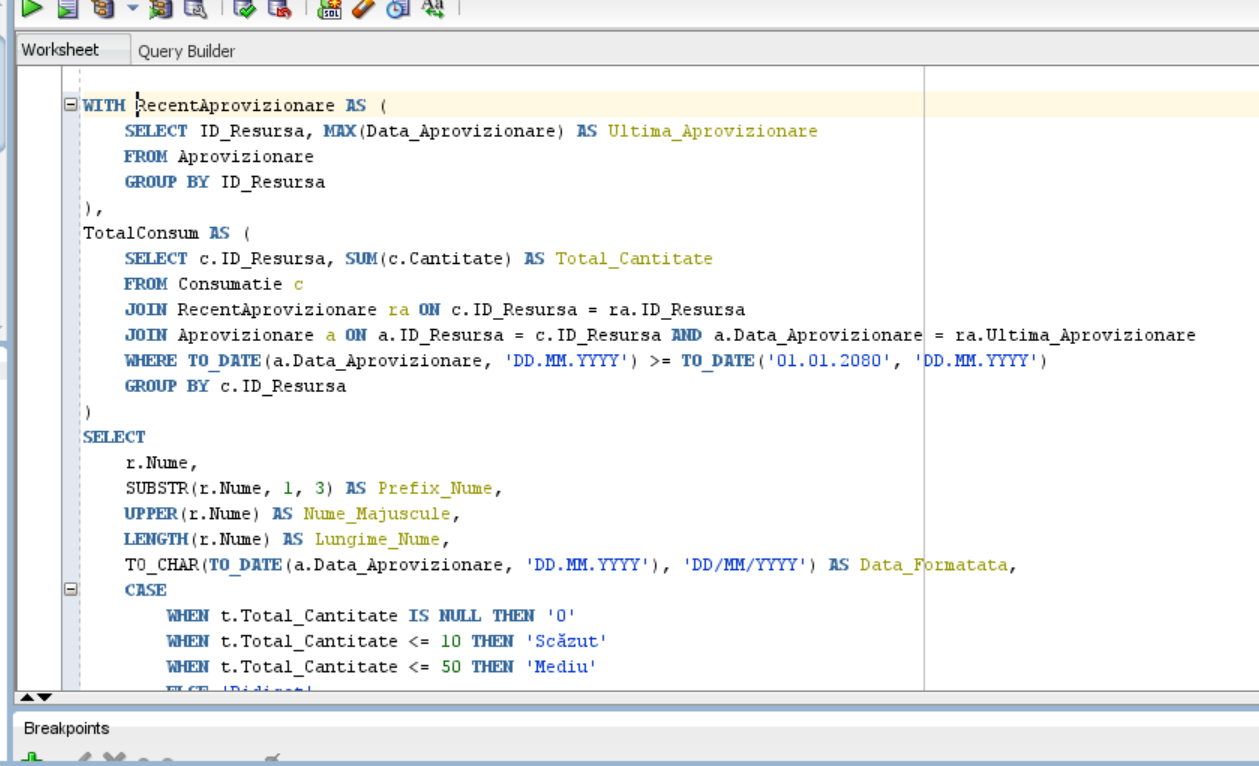
TotalConsum t ON r.ID\_Resursa = t.ID\_Resursa

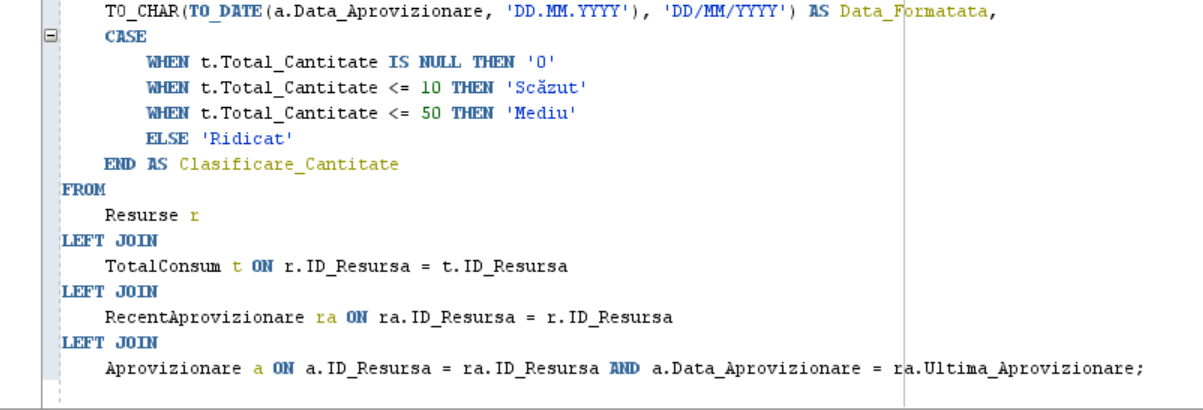
LEFT JOIN

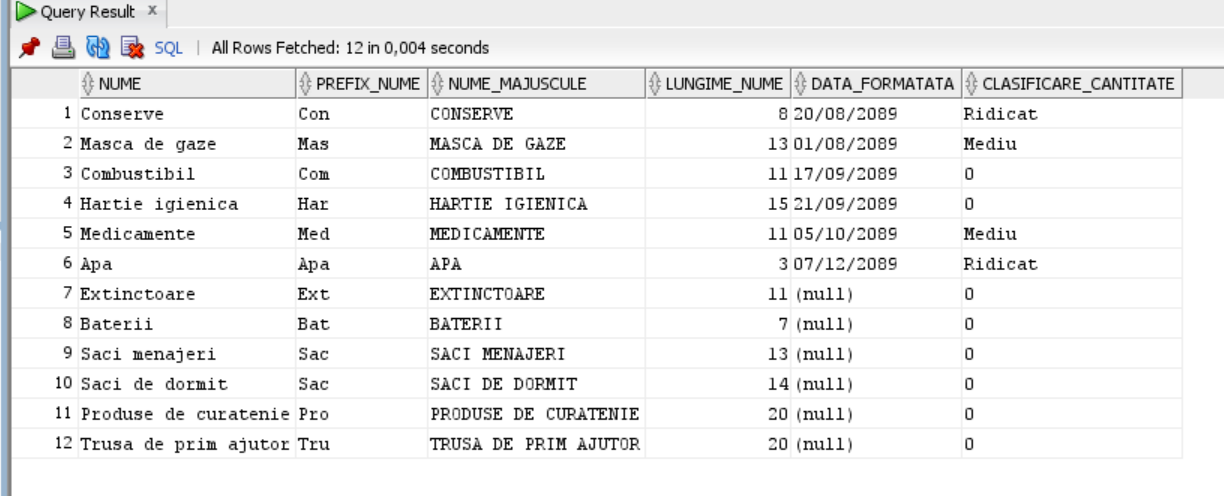
RecentAprovizionare ra ON ra.ID\_Resursa = r.ID\_Resursa

LEFT JOIN

Aprovizionare a ON a.ID\_Resursa = ra.ID\_Resursa AND a.Data\_Aprovizionare = ra.Ultima\_Aprovizionare;







13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

**Prima operație:** Actualizarea coloanei Capacitate din tabelul CAMERE pentru camerele care aparțin unui departament cu id-ul mai mic decât 3. Capacitatea va crește cu 2 oameni.

UPDATE Camere

SET Capacitate = Capacitate + 2

WHERE ID\_Departament IN (

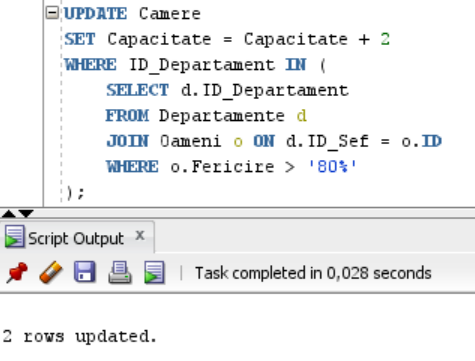
SELECT d.ID\_ Departament

FROM Departamente d

JOIN Oameni o ON d.ID\_Sef = o.ID

WHERE o.Fericire > '80%'

);



**A doua operație:** Ștergerea înregistrărilor din tabelul CONSUMAȚIE pentru consumațiile mai vechi de anul 2085 (OBS: deși putea fii scris mai simplificat, am ales această scriere pentru ilustrarea subcederii)

DELETE FROM Consumatie

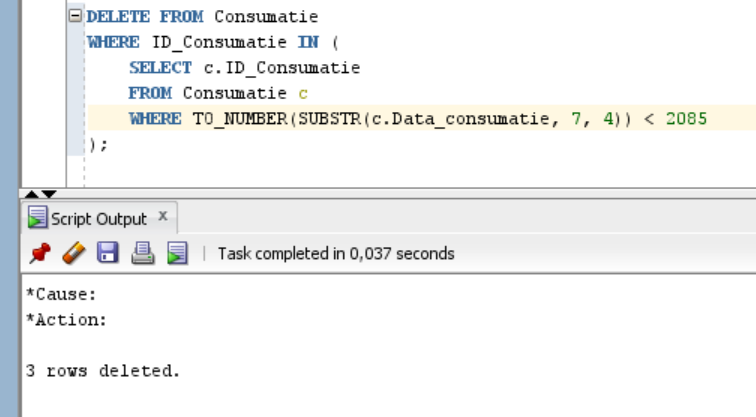
WHERE ID\_Consumatie IN (

SELECT c.ID\_Consumatie

FROM Consumatie c

WHERE TO\_NUMBER(SUBSTR(c.Data\_consumatie, 7, 4)) < 2085

);



**A treia operație:** Creșterea stoculul cu 20 de unități pentru resursele care au fost și aprovizionate și consumate măcar odată.

UPDATE Resurse r

SET r.Stoc = r.Stoc + 20

WHERE r.ID\_Resursa IN (

SELECT a.ID\_Resursa

FROM Aprovizionare a

JOIN Consumatie c ON a.ID\_Resursa = c.ID\_Resursa

GROUP BY a.ID\_Resursa

HAVING COUNT(a.ID\_Aprovizionare) > 0 AND COUNT(c.ID\_Consumatie) > 0

);

