

METODE DE OPTIMIZARE ȘI DISTRIBUIRE ÎN BAZE DE DATE

PROIECT: AGENȚIE DE TURISM

Studenți:

Chițimuș Mara-Ioana

Olteanu-Crăciunescu Ana-Maria

Văcaru Andreea-Alexandra

Grupa 505

CUPRINS

I.	MODUL ANALIZĂ	3
1.	Descrierea modelului ales și a obiectivelor aplicației	3
2.	Diagramele bazei de date OLTP inițiale, care va fi utilizată ca bază pentru distribuirea datelor.....	3
3.	Descrierea modului de distribuire (numărul de server-e de baze de date din rețea)	4
4.	Argumentarea deciziei de fragmentare a relațiilor.....	5
5.	Verificarea corectitudinii fragmentărilor realizate	15
6.	Argumentarea deciziei de replicare a anumitor relații sau/și de stocare a unei relații pe o singură stație.....	16
7.	Crearea schemelor conceptuale locale (corespunzătoare bazelor de date locale).....	16
8.	Lista tuturor constrângerilor ce trebuie îndeplinite de model.....	17
9.	Formularea în limbaj natural a unei cereri SQL complexe care va folosi date din mai multe fragmente și va fi optimizată în etapa de implementare. Precizarea tehnicilor de optimizare ce ar putea fi utilizate pentru această cerere particulară (avantaje / dezavantaje de utilizare pentru o anumită tehnică)	21
II.	MODUL BACK-END	22
1.	Crearea bazelor de date și a utilizatorilor.....	22
2.	Crearea relațiilor și a fragmentelor + popularea cu date a bazelor de date.....	29
3.	Furnizarea formelor de transparenta pentru întreg modelul ales	46
4.	Asigurarea sincronizării datelor pentru relațiile replicate	64
5.	Asigurarea tuturor constrângerilor de integritate.....	69
6.	Optimizarea cererii SQL	77
III.	MODUL FRONT-END.....	81

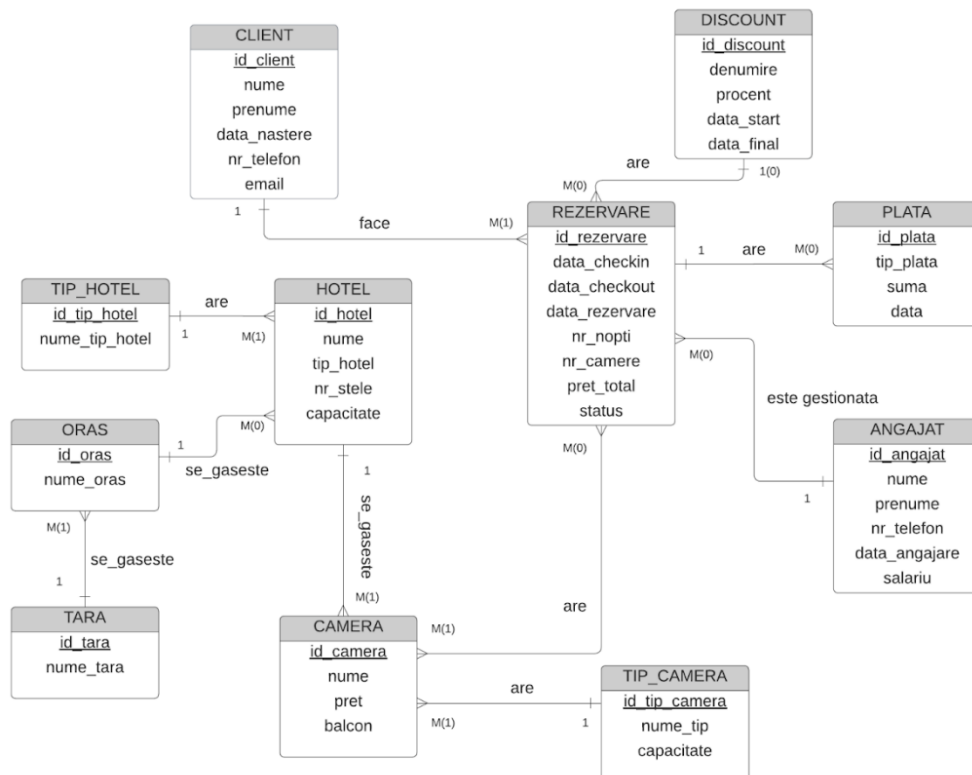
I. MODUL ANALIZĂ

1. Descrierea modelului ales și a obiectivelor aplicației

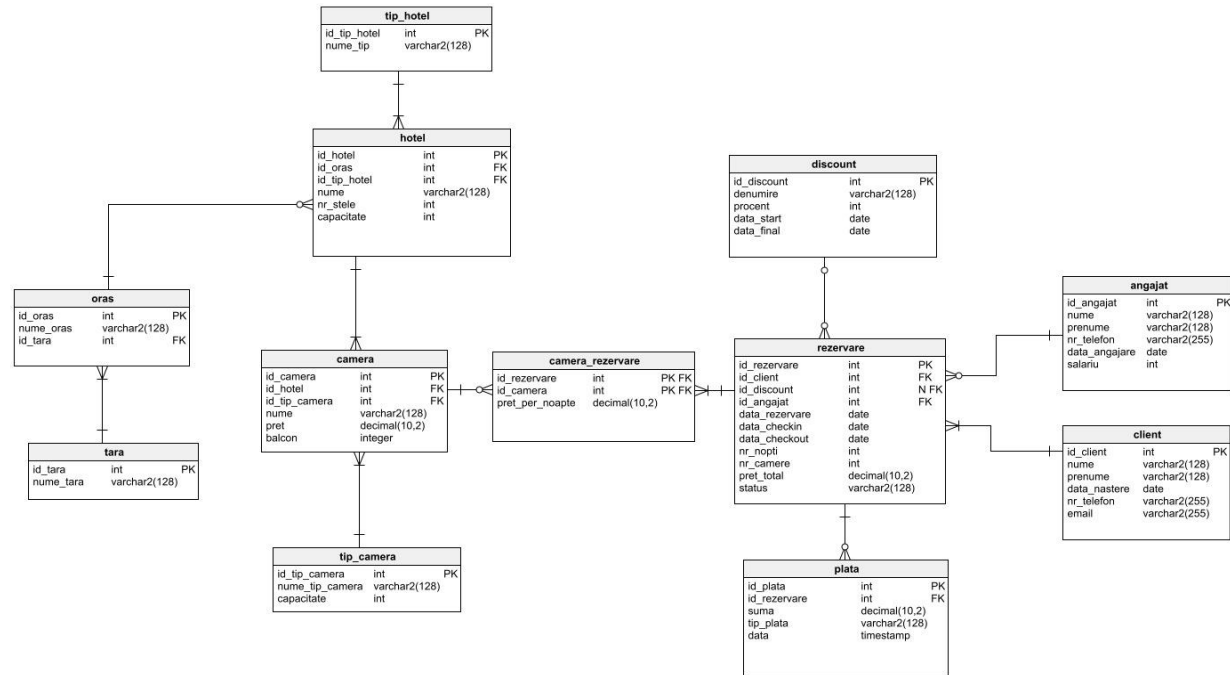
Aplicația noastră pornește de la premisa că există o bază de date inițială a unei agenții de turism ce constituie fundamentul pentru procesul de distribuire a datelor. În urma procesului de distribuire geografică a datelor se vor obține mai multe baze de date locale, două la număr, în funcție de țara hotelurilor: România sau alte țări. Se va configura o altă bază de date globală care va permite gestiunea la nivel global a tuturor datelor stocate în sistem.

2. Diagramele bazei de date OLTP inițiale, care va fi utilizată ca bază pentru distribuirea datelor

1. Diagrama entitate-relație



2. Diagrama conceptuala



3. Descrierea modului de distribuire (numărul de server-e de baze de date din rețea)

Rețeaua este formată din două servere de baze de date, fie ele server1 și server2. Pe ambele servere este creat utilizatorul MODBD_TRAVEL cu parola indicată. Pe stația server1 rezidă o bază de date al cărui nume pentru serviciul de rețea este bd_ro, iar pe stația server2 rezidă o bază de date al cărui nume pentru serviciul de rețea este bd_other. Între cele două servere de baze de date sunt configurate servicii de rețea:

- **bd_other**, cu direcția legăturii de la stația server1 la stația server2
- **bd_ro**, cu direcția legăturii de la stația server2 la stația server1

Astfel, de pe stația server1 sunt permise comenzi la distanță la bd_other, iar de pe stația server2 sunt permise comenzi la distanță la bd_ro.

```
CREATE PUBLIC DATABASE LINK bd_ro
CONNECT TO modbd_travel
IDENTIFIED BY modbdtravel
USING '(DESCRIPTION=
      (ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
      (CONNECT_DATA=(SID=db1))
)';
```

```
CREATE PUBLIC DATABASE LINK bd_other
CONNECT TO modbd_travel
IDENTIFIED BY modbdtravel
USING '(DESCRIPTION=
      (ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
      (CONNECT_DATA=(SID=db2))
)';
```

4. Argumentarea deciziei de fragmentare a relațiilor

a. Fragmentare orizontală primară

Se consideră relația TARA. Fie p_1 și p_2 două predicate, cel mai des utilizate de aplicații:

- p_1 : nume_tara = 'Romania'
- p_2 : nume_tara != 'Romania'

$Pr = \{p_1, p_2\}$ este mulțimea de predicate simple, deoarece predicatele simple p_1 și p_2 sunt cele mai des utilizate predicate în aplicații.

- Pr este completă și minimală.
 - Deoarece există probabilități de acces egale ale fiecărei aplicații la orice tuplu care aparține oricărui fragment orizontal ce este definit pe baza acestei mulțimi, rezultă că mulțimea Pr este completă.
 - Mulțimea Pr este minimală, deoarece toate predicatele mulțimii Pr sunt relevante (un predicat simplu determină divizarea unui fragment f în fragmentele f_i și f_j , dacă există cel puțin o aplicație care să acceseze fragmentele f_i și f_j în mod diferit; acest predicat simplu va fi relevant pentru fragmentare).

- Prin interogarea tabelii TARA se deduce că $\text{nume_tara} \in \{\text{'Romania'}, \text{'!Romania'}\}$
- Se construiește mulțimea de implicații I:
 - i1: $(\text{nume_tara} = \text{'Romania'}) \Rightarrow \neg(\text{region_name} \neq \text{'Romania'})$
 - i2: $\neg(\text{nume_tara} = \text{'Romania'}) \Rightarrow (\text{nume_tara} \neq \text{'Romania'})$
- Pe baza multimii de predicate simple Pr, folosind combinatii boolene, se poate forma multimea M de predicate compuse:

m1: $\text{nume_tara} = \text{'Romania'} \wedge \text{nume_tara} \neq \text{'Romania'}$

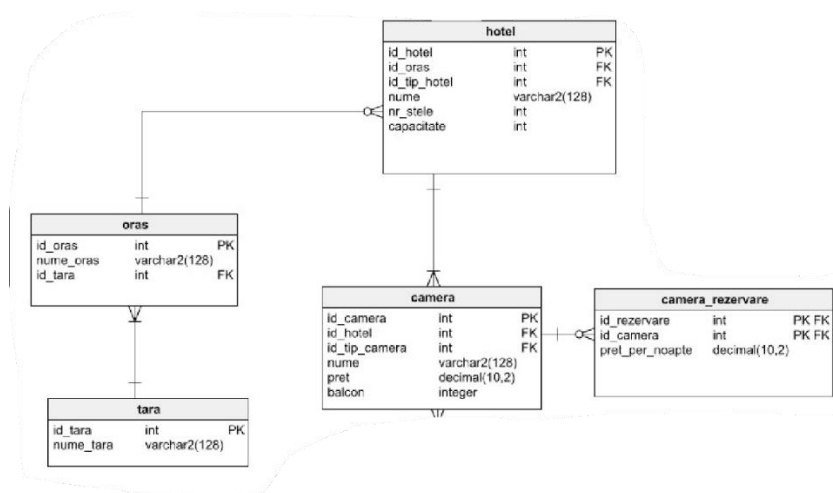
m2: $\text{nume_tara} = \text{'Romania'} \wedge \neg(\text{nume_tara} \neq \text{'Romania'})$

m3: $\neg(\text{nume_tara} = \text{'Romania'}) \wedge \text{nume_tara} \neq \text{'Romania'}$

m4: $\neg(\text{nume_tara} = \text{'Romania'}) \wedge \neg(\text{nume_tara} \neq \text{'Romania'})$
- Se observa ca:
 - predicatele m1 și m4 nu au sens în raport cu implicațiile mulțimii I și de aceea trebuie eliminate din mulțimea M
 - $m2 = p1$ si $m3 = p2$
- Deci multimea de predicate care determina fragmentele este $M = Pr = \{p1, p2\}$

b. Fragmentare orizontală derivată

Se considera următoarele relații:



Aceste relatii vor fi fragmentate pornind de la predicatele:

- p1: nume_tara = 'Romania'
- p2: nume_tara != 'Romania'

Fie legatura L dintre relațiile TARA și ORAS

owner(L) = TARA

member(L) = ORAS

In functie de tara, putem grupa orasele in:

- Orașe din România
- Orase din alte tari

$$\text{ORAS_RO} = \text{ORAS} \bowtie \text{TARA_RO}$$
$$\text{ORAS_OTHER} = \text{ORAS} \bowtie \text{TARA_OTHER}$$

In functie de tara, putem grupa hotelurile în:

- Hoteluri din orașe din România
- Hoteluri din orașe din alte tari

$$\text{HOTEL_RO} = \text{HOTEL} \bowtie \text{ORAS_RO}$$
$$\text{HOTEL_OTHER} = \text{HOTEL} \bowtie \text{ORAS_OTHER}$$

In functie de tara, putem grupa camerele in:

- Camere din hoteluri din orașe din România
- Camere din hoteluri din orașe din alte tari

$$\text{CAMERA_RO} = \text{CAMERA} \bowtie \text{HOTEL_RO}$$
$$\text{CAMERA_OTHER} = \text{CAMERA} \bowtie \text{HOTEL_OTHER}$$

Fragmentele vor fi stocate astfel:

- pe statia 1 (bd_ro):

ORAS_RO, HOTEL_RO, CAMERA_RO, CAMERA_REZERVARE_RO

- pe statia 2 (bd_other):

ORAS_OTHER, HOTEL_OTHER, CAMERA_OTHER,
CAMERA_REZERVARE_OTHER

c. Fragmentare verticală

i. Aplicarea algoritmului de fragmentare verticală (exemplificarea pașilor algoritmului pe baza unor date ipotetice)

ii. Obținerea fragmentelor verticale

- Pentru fragmentarea verticală, vom presupune că pe stăția bd_other (S2) sunt păstrate informațiile sensibile. Fragmentarea se va face între stățiile bd_ro (S1) și bd_other (S2).

- **Considerăm relația:**

ANGAJAT(id_angajat, nume, prenume, nr_telefon, data_angajare, salariu)

- Notatii:

A1 = nume

A2 = prenume

A3 = nr_telefon

A4 = data_angajare

A5 = salariu

- **Cereri relevante pentru fragmentare**

- q1, care găsește numele și prenumele angajaților - A1 și A2

SELECT nume, prenume

FROM angajat;

- q2, găsește numele și prenumele persoanelor angajate în anul 2022 - A1, A2, A4

SELECT nume, prenume

FROM angajat

WHERE EXTRACT (YEAR FROM data_angajare) = '2022';

- q3, găsește numele și prenumele angajaților cu salariul mai mare de 10 000 - A1, A2 și A5

SELECT nume, prenume

FROM angajat

WHERE salariu > 10 000;

- q4, gaseste numele si prenumele angajatilor cu salariul mai mic de 5000 - A1, A2, A5

SELECT nume, prenume

FROM angajat

WHERE salariu < 5 000;

Matricea utilizarii atributelor : VA

VA	A1	A2	A3	A4	A5
q1	1	1	0	0	0
q2	1	1	0	1	0
q3	1	1	0	0	1
q4	1	1	0	0	1

- Reteaua este formata din 2 statii: S1 si S2
- $nr_acc_l(qk) = 1$, oricare ar fi $k=1,2,3,4$ si oricare $l = 1, 2$

Frecvențele de acces ale aplicatiilor sunt:

	Statia 1	Statia 2
fr_acc_l(q1)	0	20
fr_acc_l(q2)	20	3
fr_acc_l(q3)	10	2
fr_acc_(q4)	15	0

Matricea afinității atributelor:

$$af(A_i, A_j) = \sum_{k \in K} \sum_{\forall S_l} nr_{acc_l}(q_k) f_{acc_l}(q_k)$$

$$K = \{k \mid ref(q_k, A_i) = 1 \} \wedge ref(q_k, A_j) = 1\}$$

$$af(A_1, A_1) = fr_{acc_1}(q1) + fr_{acc_2}(q1) + fr_{acc_1}(q2) + fr_{acc_2}(q2) + fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 0 + 20 + 3 + 10 + 2 + 15 + 0 = 50$$

$$af(A_1, A_2) = fr_{acc_1}(q1) + fr_{acc_2}(q1) + fr_{acc_1}(q2) + fr_{acc_2}(q2) + fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 0 + 20 + 3 + 10 + 2 + 15 + 0 = 50$$

$$af(A_1, A_3) = 0$$

$$af(A_1, A_4) = fr_{acc_1}(q2) + fr_{acc_2}(q2) = 20 + 3 = 23$$

$$af(A_1, A_5) = fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 10 + 2 + 15 + 0 = 27$$

$$af(A_2, A_2) = fr_{acc_1}(q1) + fr_{acc_2}(q1) + fr_{acc_1}(q2) + fr_{acc_2}(q2) + fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 0 + 20 + 3 + 10 + 2 + 15 + 0 = 50$$

$$af(A_2, A_3) = 0$$

$$af(A_2, A_4) = fr_{acc_1}(q2) + fr_{acc_2}(q2) = 20 + 3 = 23$$

$$af(A_2, A_5) = fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 10 + 2 + 15 + 0 = 27$$

$$af(A_3, A_3) = 0$$

$$af(A_3, A_4) = 0$$

$$af(A_3, A_5) = 0$$

$$af(A_4, A_4) = fr_{acc_1}(q2) + fr_{acc_2}(q2) = 20 + 3 = 23$$

$$af(A_4, A_5) = 0$$

$$af(A_5, A_5) = fr_{acc_1}(q3) + fr_{acc_2}(q3) + fr_{acc_1}(q4) + fr_{acc_2}(q4) = 10 + 2 + 15 + 0 = 27$$

AA	A1	A2	A3	A4	A5
A1	50	50	0	23	27
A2	50	50	0	23	27
A3	0	0	0	0	0
A4	23	23	0	23	0
A5	27	27	0	0	27

Pasul 2: Matricea legăturilor de afinitate

Etapa 1 - initializarea

Plasam la început attributele 1 si 2 (pe primele două poziții)

Pasul următor: decizie atributul A3

Calculam contribuția la afinitatea atributelor

$$leg(A_1, A_2) = 50 * 50 + 50 * 50 + 0 * 0 + 23 * 23 + 27 * 27 = 6\ 258$$

$$cont(A_x, A_k, A_z) = 2leg(A_x, A_k) + 2leg(A_k, A_z) - 2leg(A_x, A_z)$$

$$cont(A_0, A_3, A_1) = 2leg(A_0, A_3) + 2leg(A_3, A_1) - 2leg(A_0, A_1) = 2 * 0 + 2 * 0 - 2 * 0 = 0$$

$$cont(A_1, A_3, A_2) = 2leg(A_1, A_3) + 2leg(A_3, A_2) - 2leg(A_1, A_2) = 2 * 0 + 2 * 0 - 2 * 6258 \\ = -12\ 516$$

$$cont(A_2, A_3, A_0) = 2leg(A_2, A_3) + 2leg(A_3, A_0) - 2leg(A_2, A_0) = 2 * 0 + 2 * 0 - 2 * 6258 = 0$$

$$cont(A_0, A_4, A_3) = 2leg(A_0, A_4) + 2leg(A_4, A_3) - 2leg(A_0, A_3) = 2 * 0 + 2 * 0 - 2 * 0 = 0$$

$$cont(A_3, A_4, A_1) = 2leg(A_3, A_4) + 2leg(A_4, A_1) - 2leg(A_3, A_1) \\ = 2 * 0 + 2 * (50 * 23 + 50 * 23 + 0 * 0 + 23 * 23 + 27 * 0) - 2 * 0 = 2 * 2829 \\ = 5\ 658$$

$$cont(A_1, A_4, A_2) = 2leg(A_1, A_4) + 2leg(A_4, A_2) - 2leg(A_1, A_2) \\ = 2 * 2829 + 2 * (50 * 23 + 50 * 23 + 0 * 0 + 23 * 23 + 27 * 0) - 2 * 6258 \\ = 5658 + 5658 - 12\ 516 = -1\ 200$$

$$cont(A_2, A_4, A_0) = 2leg(A_2, A_4) + 2leg(A_4, A_0) - 2leg(A_2, A_0) = 2 * 2829 + 2 * 0 - 2 * 0 \\ = 5658$$

$$cont(A_0, A_5, A_3) = 2leg(A_0, A_5) + 2leg(A_5, A_3) - 2leg(A_0, A_3) = 2 * 0 + 2 * 0 - 2 * 0 = 0$$

$$cont(A_3, A_5, A_4) = 2leg(A_3, A_5) + 2leg(A_5, A_4) - 2leg(A_3, A_4) = 2 * 0 + 2 * 1242 - 2 * 0 \\ = 2484$$

$$cont(A_4, A_5, A_1) = 2leg(A_4, A_5) + 2leg(A_5, A_1) - 2leg(A_4, A_1) \\ = 2 * 1242 + 2 * 3429 - 2 * 2829 = 2482 + 6858 - 5658 = 3\ 682$$

$$cont(A_1, A_5, A_2) = 2leg(A_1, A_5) + 2leg(A_5, A_2) - 2leg(A_1, A_2) \\ = 2 * 3429 + 2 * 3429 - 2 * 6258 = 6\ 858 + 6\ 858 - 12\ 516 = 1\ 200$$

$$cont(A_2, A_5, A_0) = 2leg(A_2, A_5) + 2leg(A_5, A_0) - 2leg(A_2, A_0) = 2 * 3429 + 2 * 0 - 2 * 0 \\ = 6\ 858$$

La	A3	A4	A1	A2	A5
L1	0	23	50	50	27
L2	0	23	50	50	27
L3	0	0	0	0	0
L4	0	23	23	23	0
L5	0	0	27	27	27

Pentru ca matricea sa fie simetrică, vom permuta L1 cu L3 si L2 cu L4, rezultand:

La	A3	A4	A1	A2	A5
L3	0	0	0	0	0
L4	0	23	23	23	0
L1	0	23	50	50	27
L2	0	23	50	50	27
L5	0	0	27	27	27

Mă plimb pe diagonală și verific dacă punctul la care am ajuns e optim sau nu pentru divizare.

$$Q = \{q1, q2, q3, q4\}$$

$$n = 1$$

TQ1 = \emptyset (aplicațiile care accesează doar atributul A3)

BQ1 = \emptyset (aplicațiile care accesează doar atributele A4, A1, A2, A5)

OQ1 = {q1, q2, q3, q4}

CTQ1 = 0

CBQ1 = 0

$$COQ1 = fr_{acc_1(q1)} + fr_{acc_2(q1)} + fr_{acc_1(q2)} + fr_{acc_2(q2)} + fr_{acc_1(q3)} + fr_{acc_2(q3)} + fr_{acc_1(q4)} + fr_{acc_2(q4)} = 0 + 20 + 20 + 3 + 10 + 2 + 15 + 0 = 70$$

$$z1 = CTQ1 * CBQ1 - COQ1^2 = -4\ 900$$

$$n = 2$$

TQ2 = \emptyset (aplicațiile care accesează doar atributele A3 și A4)

BQ2 = {q3, q4} (aplicațiile care accesează doar atributele A1, A2, A5)

OQ2 = {q1, q2}

CTQ2 = 0

$$CBQ2 = fr_{acc_1(q3)} + fr_{acc_2(q3)} + fr_{acc_1(q4)} + fr_{acc_2(q4)} = 27$$

$$COQ2 = fr_{acc_1(q1)} + fr_{acc_2(q1)} + fr_{acc_1(q2)} + fr_{acc_2(q2)} = 43$$

$$z2 = CTQ2 * CBQ2 - COQ2^2 = -1\ 849$$

$$n = 3$$

TQ3 = \emptyset (aplicațiile care accesează doar atributele A3, A4 și A1)

BQ3 = \emptyset (aplicațiile care accesează doar atributele A2, A5)

OQ3 = {q1, q2, q3, q4}

CTQ3 = 0

CBQ3 = 0

$$COQ3 = fr_{acc_1(q1)} + fr_{acc_2(q1)} + fr_{acc_1(q2)} + fr_{acc_2(q2)} + fr_{acc_1(q3)} + fr_{acc_2(q3)} + fr_{acc_1(q4)} + fr_{acc_2(q4)} = 0 + 20 + 20 + 3 + 10 + 2 + 15 + 0 = 70$$

$$z3 = CTQ3 * CBQ3 - COQ3^2 = -4\ 900$$

$$n = 4$$

TQ4 = \emptyset (aplicațiile care accesează doar atributele A3, A4, A1 și A2)

BQ4 = \emptyset (aplicațiile care accesează doar atributele A5)

OQ4 = {q1, q2, q3, q4}

$$CTQ4 = 0$$

$$CBQ4 = 0$$

$$COQ4 = fr_{acc_1(q1)} + fr_{acc_2(q1)} + fr_{acc_1(q2)} + fr_{acc_2(q2)} + fr_{acc_1(q3)} + fr_{acc_2(q3)} + fr_{acc_1(q4)} + fr_{acc_2(q4)} = 0 + 20 + 20 + 3 + 10 + 2 + 15 + 0 = 70$$

$$z4 = CTQ4 * CBQ4 - COQ4^2 = -4\,900$$

Rezultă că optim este $z = z2 = -1\,849$, deci punctul de partiționare este 2. A0 este cheia primară și va fi conținută de ambele fragmente. Deci cele două fragmente sunt: {A0 , A3, A4} și {A0, A1, A2, A5}.

A0 = id_angajat

A1 = nume

A2 = prenume

A3 = nr_telefon

A4 = data_angajare

A5 = salariu

ANGAJAT_1(id_angajat, nr_telefon, data_angajare)

ANGAJAT_2(id_angajat, nume, prenume, salariu)

Pentru implementare am ales să stocăm pe stația bd_other datele sensibile (id_angajat, data_angajare, salariu, telefon), iar pe stația bd_ro celelalte date: nume și prenume, împreună cu id_angajat.

5. Verificarea corectitudinii fragmentărilor realizate

a. Fragmentare orizontală primară

- Fragmentarea este corectă dacă sunt îndeplinite regulile de completitudine, reconstrucție și disjuncție.
- Completitudinea: M este completă, deci proprietatea de completitudine este asigurată.
- Reconstrucția:
 - $TARA = TARA_RO \cup TARA_OTHER$
- Disjuncția:
 - $TARA_RO \cap TARA_OTHER = \emptyset$

b. Fragmentare orizontală derivată

- ORAS

Completitudinea: Este asigurată, deoarece fiecărui tuplu din cele două fragmente ale relației ORAS îi corespunde un tuplu din relația TARA.

Reconstrucția:

- $ORAS = ORAS_RO \cup ORAS_OTHER$

Disjuncția: Este asigurată, deoarece graful ce reprezintă diagrama relațiilor (TARA, ORAS) și a legăturilor dintre acestea este un graf simplu (există o singură legătură ce pleacă sau vine pentru orice fragment).

- HOTEL

Completitudinea: Este asigurată, deoarece fiecărui tuplu din cele două fragmente ale relației HOTEL îi corespunde un tuplu din relația ORAS.

Reconstrucția:

- $HOTEL = HOTEL_RO \cup HOTEL_OTHER$

Disjuncția: Este asigurată, deoarece graful ce reprezintă diagrama relațiilor (ORAS, HOTEL) și a legăturilor dintre acestea este un graf simplu (există o singură legătură ce pleacă sau vine pentru orice fragment).

- CAMERA, CAMERA_REZERVARE similar cu HOTEL

6. Argumentarea deciziei de replicare a anumitor relații sau/și de stocare a unei relații pe o singură stație

- Relația discount este replicată, deoarece, discount-urile sunt definite pe o perioadă de timp și sunt aplicate rezervărilor indiferent de zonă
- Deoarece tabela tip_hotel conține un nomenclator cu 5 tipuri de hoteluri: Hotel, Pensiune, Resort, Vilă și Motel, fiind o tabelă pe care nu o modificăm frecvent, aceasta va fi replicată
- Relația client este o relație replicată, deoarece un client poate face rezervare atât pentru un hotel din România, cât și pentru un hotel din altă țară
- Deoarece tabela tip_camera conține un nomenclator cu 4 tipuri de camere: Single, Double, Triple, Apartament, fiind o tabelă pe care nu o modificăm frecvent, aceasta va fi replicată

7. Crearea schemelor conceptuale locale (corespunzătoare bazelor de date locale)

Diagrama conceptuala statia bd_ro

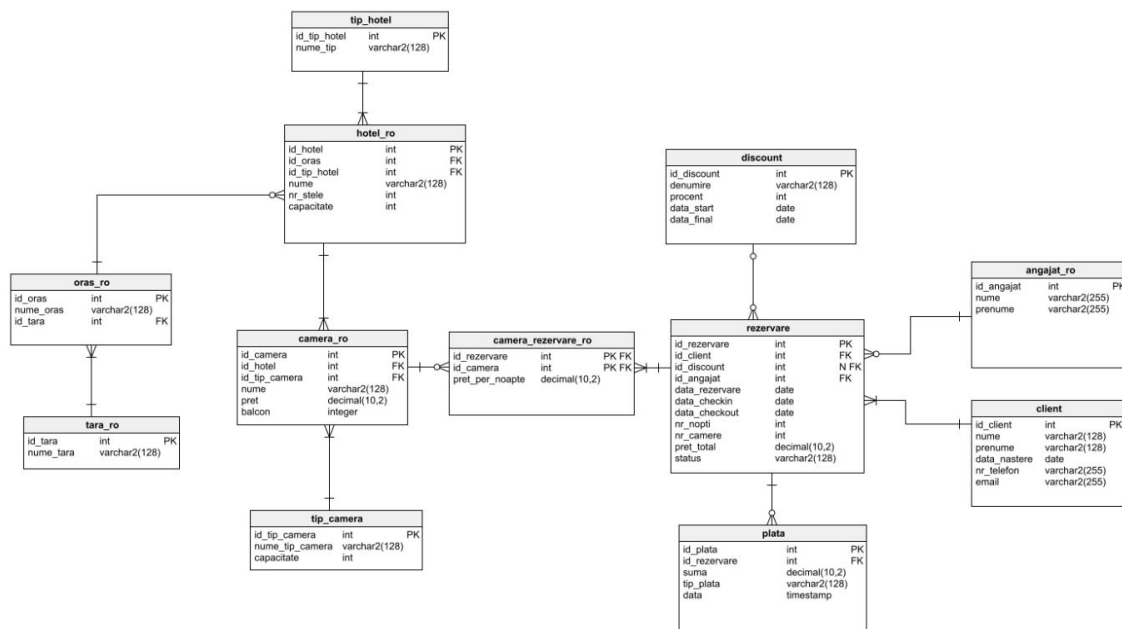
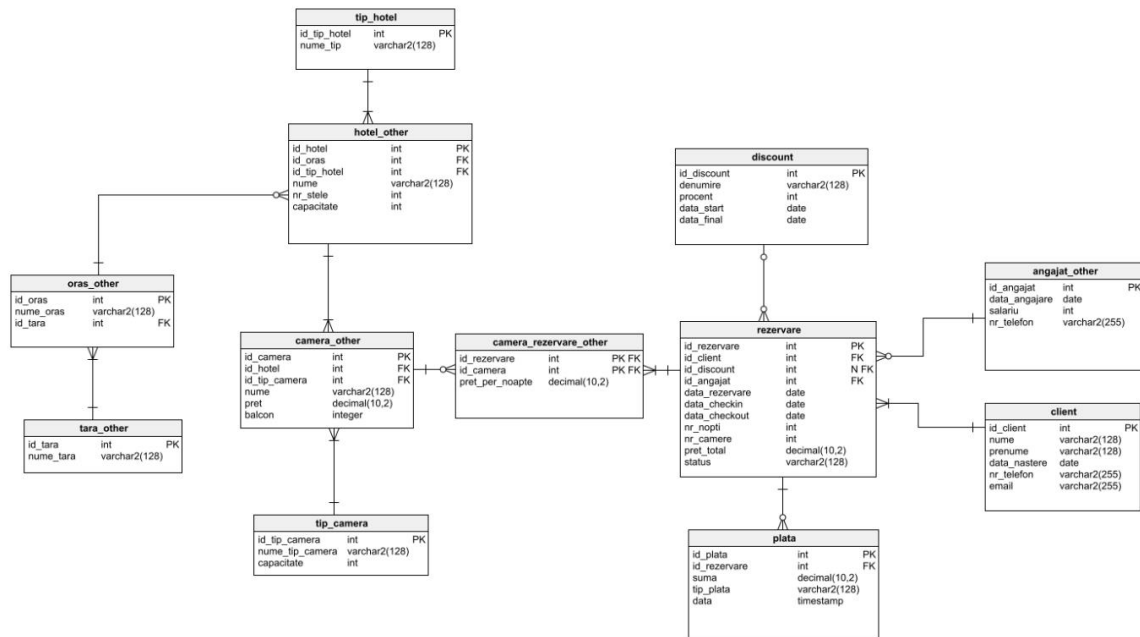


Diagrama conceptuala statia bd_other



8. Lista tuturor constrângerilor ce trebuie îndeplinite de model

a) Unicitate

i. unicitate locală

La nivel local, vom asigura unicitatea prin adaugarea constrangerilor de unicitate pentru:

Statie	Nume tabel	Coloane	Constrangere
bd_ro	oras_ro	nume_oras	UNIQUE
bd_other	oras_other	nume_oras	UNIQUE
bd_other	angajat_other	nr_telefon	UNIQUE

ii. unicitate globală pe fragmente orizontale

La nivel global, vom asigura unicitatea pe fragmente orizontale pentru relatia oras utilizand triggerii:

- Trigg_oras_ro prin care se verifica inainte de a insera sau modifica o noua inregistrare in tabela oras_ro daca exista o inregistrare avand aceleasi valori pentru nume_oras in tabela distanta oras_other;
- Trigg_oras_other prin care se verifica inainte de a insera sau modifica o noua inregistrare in tabela oras_other daca exista o inregistrare avand aceleasi valori pentru nume_oras in tabela distanta oras_ro. Se vor verifica cei doi triggeri prin inserarea in tabelele oras_ro, respectiv oras_other.

iii. unicitate globală în cazul în care trebuie să fie unică o combinație de coloane care se găsesc în fragmente verticale diferite, specificându-se argumentele care au stat la baza acestei decizii din punct de vedere al optimizării

- La nivel global, vom asigura unicitatea pe fragmentul vertical pentru relatia angajat prin crearea unui trigger numit trigg_angajat_unique care va verifica pentru fiecare inserare in view-ul angajat daca exista sau nu un angajat care sa aiba deja acelasi nume si nr_telefon in tabelele fragmentate angajat_ro si angajat_other.

b) Cheie primară (la nivel local/global)

- La nivel local

Nume tabel	Coloane	Constrangere
tara_ro / tara_other	id_tara	PRIMARY KEY
oras_ro / oras_other	id_oras	PRIMARY KEY
hotel_ro / hotel_other	id_hotel	PRIMARY KEY
camera_ro / camera_other	id_camera	PRIMARY KEY

camera_rezervare_ro/ camera_rezervare_other	(id_rezervare, id_camera)	PRIMARY KEY
angajat_ro / angajat_other	id_angajat	PRIMARY KEY

- La nivel global

Se vor asigura constrangerile de cheie primara pentru relatiile tara, oras, hotel, camera, camera_rezervare si angajat (fragmentate orizontal si vertical) prin utilizarea secventelor.

- Statia bd_ro

Nume tabel	Nume Secventa	Detalii secventa
tara_ro	sec_tara_ro	Secventa care are pasul de incrementare 2, va incepe de la 7000 si nu va cicla
oras_ro	sec_oras_ro	Secventa care are pasul de incrementare 2, va incepe de la 10000 si nu va cicla
hotel_ro	sec_hotel_ro	Secventa care are pasul de incrementare 2, va incepe de la 1000 si nu va cicla
camera_ro	sec_camera_ro	Secventa care are pasul de incrementare 2, va incepe de la 1000 si nu va cicla
angajat_ro	sec_angajat_ro	Secventa care are pasul de incrementare 2, va incepe de la 1000 si nu va cicla

- Statia bd_other

Nume tabel	Nume Secventa	Detalii secventa
tara_other	sec_tara_other	Secventa care are pasul de incrementare 2, va incepe de la 7001 si nu va cicla
oras_other	sec_oras_other	Secventa care are pasul de incrementare 2, va incepe de la 10001 si nu va cicla
hotel_other	sec_hotel_other	Secventa care are pasul de incrementare 2, va incepe de la 1001 si nu va cicla

camera_other	sec_camera_other	Secventa care are pasul de incrementare 2, va incepe de la 1001 si nu va cicla
angajat_other	sec_angajat_other	Secventa care are pasul de incrementare 2, va incepe de la 1001 si nu va cicla

c) Cheie externă (la nivel local și pentru relații stocate în baze de date diferite)

La nivel local, vor si asigurate constrangerile de cheie externa, atat pentru statia bd_ro, cat si pentru statia bd_other.

1. Pentru fragmentele tara_ro/tara_other si oras_ro/oras_other constrângerea de cheie externă se realizează local, deoarece ambele fragmente sunt situate pe aceeași stație.
2. Pentru fragmentele oras_ro/oras_other si hotel_ro/hotel_other constrângerea de cheie externă se realizează local, deoarece ambele fragmente sunt situate pe aceeași stație.
3. Pentru fragmentele hotel_ro/hotel_other si camera_ro/camera_other constrângerea de cheie externă se realizează local, deoarece ambele fragmente sunt situate pe aceeași stație.

d) Validare (la nivel local și pentru relații stocate în baze de date diferite)

Validarea se realizeaza la nivel local fiecarui fragment implicat, deoarece atributele (care trebuie sa indeplineasca conditia) fac parte din acelasi fragment:

Statie	Nume tabel	Coloane	Constrangere
bd_ro/ bd_other	hotel_ro / hotel_other	capacitate	CHECK (>0)
		nr_stele	CHECK(IN (1,2,3,4,5))
Bd_ro / bd_other	camera_ro/ camera_other	pret	CHECK(>0)
		balcon	CHECK(IN (0,1))
bd_ro	angajat_ro	nume	CHECK (NOT LIKE '%[0- 9]%')
		prenume	CHECK (NOT LIKE '%[0- 9]%')

bd_other	angajat_other	data_angajare	CHECK(data_angajare <SYSDATE)
		salariu	CHECK(>0)

9. Formularea în limbaj natural a unei cereri SQL complexe care va folosi date din mai multe fragmente și va fi optimizată în etapa de implementare. Precizarea tehnicilor de optimizare ce ar putea fi utilizate pentru această cerere particulară (avantaje / dezavantaje de utilizare pentru o anumită tehnică)

Sa se afiseze rezervările care au fost efectuate de un angajat cu salariul de cel puțin 5000 de lei. Pentru aceasta cerinta se pot aplica optimizari bazate pe regula si optimizari bazate pe cost. În general, optimizarea bazată pe cost este considerată mai avansată și mai precisă decât optimizarea bazată pe regulă, deoarece ia în considerare mai mulți factori în procesul de optimizare. Cu toate acestea, optimizarea bazată pe regulă poate fi utilă în anumite situații, cum ar fi atunci când statisticile despre datele din baza de date sunt incomplete sau inexacte.

II. MODUL BACK-END

1. Crearea bazelor de date si a utilizatorilor

```
-----  
--USER SYS PE DB1  
-----
```

```
--1. Crearea bazelor de date si a utilizatorilor
```

```
create user modbd_travel identified by modbdtravel;  
grant create session to modbd_travel;
```

```
create user modbd_centralizat identified by modbdcentralizat;  
grant create session to modbd_centralizat;
```

```
create user modbd_global identified by modbdglobal;  
grant create session to modbd_global;  
grant create view to modbd_global;  
grant create trigger to modbd_global;
```

```
--grant-uri pentru 'CREATE DATABASE LINK', 'CREATE PUBLIC DATABASE LINK'  
grant create database link to modbd_travel;  
grant create public database link to modbd_travel;
```

```
GRANT UNLIMITED TABLESPACE TO modbd_travel;  
grant create table to modbd_travel;  
grant create trigger to modbd_travel;  
grant create sequence to modbd_travel;
```

```
grant all privileges to modbd_centralizat;
```

```
-----  
--BD_CENTRALIZAT  
-----
```

```
-----  
-- TABELE DIN modbd_centralizat  
-----
```

```
CREATE TABLE modbd_centralizat.angajat (  
  id_angajat int NOT NULL,  
  nume varchar2(128) NOT NULL,  
  prenume varchar2(128) NOT NULL,
```

```

nr_telefon varchar2(255) NOT NULL,
data_angajare date NOT NULL,
salariu int NOT NULL,
CONSTRAINT angajat_pk PRIMARY KEY (id_angajat)

);

CREATE TABLE modbd_centralizat.client (
  id_client int NOT NULL,
  nume varchar2(128) NOT NULL,
  prenume varchar2(128) NOT NULL,
  data_nastere date NOT NULL,
  nr_telefon varchar2(255) NOT NULL,
  email varchar2(255) NOT NULL,
  CONSTRAINT client_pk PRIMARY KEY (id_client)
);

CREATE TABLE modbd_centralizat.discount (
  id_discount int NOT NULL,
  denumire varchar2(128) NOT NULL,
  procent int NOT NULL,
  data_start date NOT NULL,
  data_final date NOT NULL,
  CONSTRAINT discount_pk PRIMARY KEY (id_discount)
);

CREATE TABLE modbd_centralizat.tara (
  id_tara int NOT NULL,
  nume_tara varchar2(128) NOT NULL,
  CONSTRAINT country_ak_1 UNIQUE (nume_tara),
  CONSTRAINT tara_pk PRIMARY KEY (id_tara)
);

CREATE TABLE modbd_centralizat.oras (
  id_oras int NOT NULL,
  nume_oras varchar2(128) NOT NULL,
  id_tara int NOT NULL,
  CONSTRAINT city_ak_1 UNIQUE (nume_oras, id_tara),
  CONSTRAINT oras_pk PRIMARY KEY (id_oras)
);

CREATE TABLE modbd_centralizat.tip_hotel (
  id_tip_hotel int NOT NULL,
  nume_tip varchar2(128) NOT NULL,
  CONSTRAINT room_type_ak_1 UNIQUE (nume_tip),
  CONSTRAINT tip_hotel_pk PRIMARY KEY (id_tip_hotel)

```

```
);
```

```
CREATE TABLE modbd_centralizat.hotel (  
    id_hotel int NOT NULL,  
    id_oras int NOT NULL,  
    id_tip_hotel int NOT NULL,  
    nume varchar2(128) NOT NULL,  
    nr_stele int NOT NULL,  
    capacitate int NOT NULL,  
    CONSTRAINT hotel_ak_1 UNIQUE (nume),  
    CONSTRAINT hotel_pk PRIMARY KEY (id_hotel)  
);
```

```
CREATE TABLE modbd_centralizat.tip_camera (  
    id_tip_camera int NOT NULL,  
    nume_tip_camera varchar2(128) NOT NULL,  
    capacitate int NOT NULL,  
    CONSTRAINT tip_camera_pk PRIMARY KEY (id_tip_camera)  
);
```

```
CREATE TABLE modbd_centralizat.camera (  
    id_camera int NOT NULL,  
    id_hotel int NOT NULL,  
    id_tip_camera int NOT NULL,  
    nume varchar2(128) NOT NULL,  
    pret decimal(10,2) NOT NULL,  
    balcon integer NOT NULL,  
    CONSTRAINT room_ak_1 UNIQUE (nume, id_hotel),  
    CONSTRAINT camera_pk PRIMARY KEY (id_camera)  
);
```

```
CREATE TABLE modbd_centralizat.rezervare (  
    id_rezervare int NOT NULL,  
    id_client int NOT NULL,  
    id_discount int NULL,  
    id_angajat int NOT NULL,  
    data_rezervare date NOT NULL,  
    data_checkin date NOT NULL,  
    data_checkout date NOT NULL,  
    nr_nopti int NOT NULL,  
    nr_camere int NOT NULL,  
    pret_total decimal(10,2) NOT NULL,  
    status varchar2(128) NOT NULL,  
    CONSTRAINT rezervare_pk PRIMARY KEY (id_rezervare)  
);
```

```
alter table modbd_centralizat.rezervare modify (nr_nopti null);
```



```

CREATE TABLE modbd_centralizat.plata (
    id_plata int NOT NULL,
    id_rezervare int NOT NULL,
    suma decimal(10,2) NOT NULL,
    tip_plata varchar2(128) NOT NULL,
    data timestamp NOT NULL,
    CONSTRAINT plata_pk PRIMARY KEY (id_plata)
);

CREATE TABLE modbd_centralizat.camera_rezervare (
    id_rezervare int NOT NULL,
    id_camera int NOT NULL,
    pret_per_noapte decimal(10,2) NOT NULL,
    CONSTRAINT camera_rezervare_pk PRIMARY KEY (id_rezervare,id_camera)
);

ALTER TABLE modbd_centralizat.oras ADD CONSTRAINT oras_tara
    FOREIGN KEY (id_tara)
    REFERENCES modbd_centralizat.tara (id_tara);

ALTER TABLE modbd_centralizat.rezervare ADD CONSTRAINT discount_rezervare
    FOREIGN KEY (id_discount)
    REFERENCES modbd_centralizat.discount (id_discount);

ALTER TABLE modbd_centralizat.hotel ADD CONSTRAINT hotel_oras
    FOREIGN KEY (id_oras)
    REFERENCES modbd_centralizat.oras (id_oras);

ALTER TABLE modbd_centralizat.hotel ADD CONSTRAINT hotel_tip_hotel
    FOREIGN KEY (id_tip_hotel)
    REFERENCES modbd_centralizat.tip_hotel (id_tip_hotel);

ALTER TABLE modbd_centralizat.plata ADD CONSTRAINT plata_rezervare
    FOREIGN KEY (id_rezervare)
    REFERENCES modbd_centralizat.rezervare (id_rezervare);

ALTER TABLE modbd_centralizat.rezervare ADD CONSTRAINT rezervare_client
    FOREIGN KEY (id_client)
    REFERENCES modbd_centralizat.client (id_client);

ALTER TABLE modbd_centralizat.rezervare ADD CONSTRAINT rezervare_angajat
    FOREIGN KEY (id_angajat)
    REFERENCES modbd_centralizat.angajat (id_angajat);

ALTER TABLE modbd_centralizat.camera ADD CONSTRAINT camera_hotel
    FOREIGN KEY (id_hotel)
    REFERENCES modbd_centralizat.hotel (id_hotel);

```

```
ALTER TABLE modbd_centralizat.camera_rezervare ADD CONSTRAINT camera_rezervare_rezervare
FOREIGN KEY (id_rezervare)
REFERENCES modbd_centralizat.rezervare (id_rezervare);
```

```
ALTER TABLE modbd_centralizat.camera_rezervare ADD CONSTRAINT camera_rezervare_camera
FOREIGN KEY (id_camera)
REFERENCES modbd_centralizat.camera (id_camera);
```

```
ALTER TABLE modbd_centralizat.camera ADD CONSTRAINT camera_tip_camera
FOREIGN KEY (id_tip_camera)
REFERENCES modbd_centralizat.tip_camera (id_tip_camera);
```

```
--grant-uri pentru modbd_travel in schema user-ului modbd_centralizat
```

```
declare
```

```
begin
```

```
FOR x IN (SELECT * FROM user_tables)
```

```
LOOP
```

```
EXECUTE IMMEDIATE 'GRANT SELECT ON ' || x.table_name || ' TO modbd_travel';
```

```
END LOOP;
```

```
end;
```

```
/
```

```
select * from user_sys_privs;
```

```
--drepturi pentru modbd_global
```

```
declare
```

```
begin
```

```
FOR x IN (SELECT * FROM user_tables)
```

```
LOOP
```

```
EXECUTE IMMEDIATE 'GRANT ALL ON ' || x.table_name || ' TO modbd_global';
```

```
END LOOP;
```

```
end;
```

```
/
```

```
-----  
--BD_RO  
-----
```

```
--1. Crearea bazelor de date si a utilizatorilor
```

```
--verificare grant-uri pentru modbd_travel in schema user-ului modbd_centralizat
```

```
select table_name from all_tables
```

```
where owner = 'MODBD_CENTRALIZAT';
```

```
CREATE PUBLIC DATABASE LINK bd_other
CONNECT TO modbd_travel
IDENTIFIED BY modbdtravel
USING '(DESCRIPTION=
      (ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
      (CONNECT_DATA=(SID=db2))
)';
```

```
desc user_tables;
```

```
desc user_tables@bd_other;
```

```
select table_name from user_tables@bd_other;
```

```
select table_name from all_tables;
```

```
desc all_tables;
```

```
desc user_sys_privs
```

```
select * from user_sys_privs;
```

```
select * from user_sys_privs
```

```
where privilege in
```

```
  ('CREATE DATABASE LINK',
```

```
  'CREATE PUBLIC DATABASE LINK',
```

```
  'CREATE SESSION')
```

```
order by 1;
```

```
-----
--USER SYS PE DB2
-----
```

```
--1. Crearea bazelor de date si a utilizatorilor
```

```
create user modbd_travel identified by modbdtravel;
```

```
grant create session to modbd_travel;
```

```
GRANT UNLIMITED TABLESPACE TO modbd_travel;
```

```
grant create table to modbd_travel;
```

```
--grant-uri pentru 'CREATE DATABASE LINK', 'CREATE PUBLIC DATABASE LINK'
```

```
grant create database link to modbd_travel;
```

```
grant create public database link to modbd_travel;
```

```
grant create trigger to modbd_travel;
```

```
grant create sequence to modbd_travel;
```

```
-----
```

--BD_OTHER

--1. Crearea bazelor de date si a utilizatorilor

```
CREATE PUBLIC DATABASE LINK bd_ro
CONNECT TO modbd_travel
IDENTIFIED BY modbdtravel
USING '(DESCRIPTION=
      (ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
      (CONNECT_DATA=(SID=db1))
)';
```

--nu functioneaza

desc user_tables@bd_ro;

--functioneaza





select table_name from user_tables@bd_ro;

--utilizatorul se afla pe cealalta baza de date, nu are acces

```
select table_name from all_tables
where owner = 'MODBD_CENTRALIZAT';
```

```
select * from user_sys_privs;
```

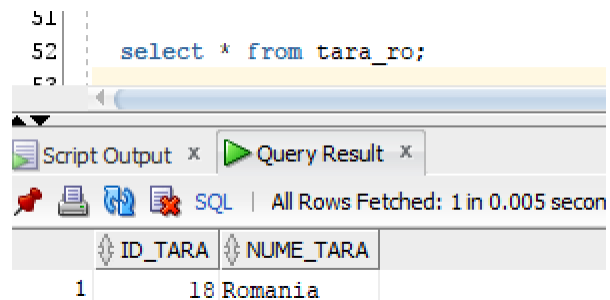
```
19 select * from user_sys_privs
20 where privilege in
21     ('CREATE DATABASE LINK',
22      'CREATE PUBLIC DATABASE LINK',
23      'CREATE SESSION')
24 order by 1;
25
```

Script Output x	Query Result x			
   	SQL All Rows Fetched: 3 in 0.002 seconds			
USERNAME	PRIVILEGE	ADMIN_OPTION	COMMON	INHERITED
1 MODBD_TRAVEL	CREATE SESSION	NO	NO	NO
2 MODBD_TRAVEL	CREATE PUBLIC DATABASE LINK	NO	NO	NO
3 MODBD_TRAVEL	CREATE DATABASE LINK	NO	NO	NO

2. Crearea relatiilor si a fragmentelor + popularea cu date a bazelor de date

1) BD_RO

```
-----  
--BD_RO  
-----  
  
-----  
--fragmentare orizontala primara TARA  
-----  
  
SELECT *  
FROM modbd_centralizat.tara;  
  
CREATE TABLE tara_ro AS  
SELECT *  
FROM modbd_centralizat.tara  
WHERE nume_tara = 'Romania';  
  
select * from tara_ro;  
  
select * from tara_ro  
union all  
select * from tara_other@bd_other;
```



51 |
52 | select * from tara_ro;
53 |

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.005 second

	ID_TARA	NUME_TARA
1	18	Romania

```

38
39 --fragmentare orizontala tabela TARA
40 CREATE TABLE tara_ro AS
41 SELECT *
42 FROM modbd_centralizat.tara
43 WHERE nume_tara = 'Romania';
44
45 select * from tara_ro
46 union all
47 select * from tara_other@bd_other;
48

```

Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 18 in 0.006 seconds

ID_TARA	NUME_TARA
1	18 Romania
2	1 Afghanistan
3	2 Algeria
4	3 American Samoa
5	4 Angola
6	5 Anguilla
7	6 Argentina
8	7 Armenia
9	8 Australia
10	9 Austria
11	10 Azerbaijan
12	11 Bahrain
13	12 Bangladesh
14	13 Belarus
15	14 Bolivia
16	15 Brazil
17	16 Brunei
18	17 Bulgaria

```

--completitudinea
select *
from modbd_centralizat.tara
minus
(select * from tara_ro
union all
select * from tara_other@bd_other);

```

```

--reconstructia
-- a=b
-- a inclus in b
select *
from modbd_centralizat.tara
minus
(select * from tara_ro
union all
select * from tara_other@bd_other);

```

```

-- b inclus in a

```

```

(select * from tara_ro
union all
select * from tara_other@bd_other)
minus
select *
from modbd_centralizat.tara;

--disjunctia
select * from tara_ro
intersect
select * from tara_other@bd_other;

```

```

40
49 --completitudinea
50 select *
51 from modbd_centralizat.tara
52 minus
53 (select * from tara_ro
54 union all
55 select * from tara_other@bd_other);
56

```

Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 0 in 0.004 seconds

ID_TARA	NUME_TARA
---------	-----------

```

57 --reconstructia
58 -- a=b
59 -- a inclus in b
60 select *
61 from modbd_centralizat.tara
62 minus
63 (select * from tara_ro
64 union all
65 select * from tara_other@bd_other);
66
67 -- b inclus in a
68
69 (select * from tara_ro
70 union all
71 select * from tara_other@bd_other)

```

Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 0 in 0.003 seconds

ID_TARA	NUME_TARA
---------	-----------

```
76 --disjunctia
77 select * from tara_ro
78 intersect
79 select * from tara_other@bd_other;
80
81
```

Script Output x | Explain Plan x | Query Result x

SQL | All Rows Fetched: 0 in 0.006 seconds

ID_TARA NUME_TARA

--fragmentare orizontala derivata

--ORAS

SELECT *
FROM modbd_centralizat.oras;

create table oras_ro
as
select a.*
from modbd_centralizat.oras a
where EXISTS
(select 1
from tara_ro b
where a.id_tara = b.id_tara
);

select * from oras_ro;

--completitudinea

select *
from modbd_centralizat.oras
minus
(select * from oras_ro
union all
select * from oras_other@bd_other);

--reconstructia

-- a=b

-- a inclus in b

select *
from modbd_centralizat.oras
minus


```
(select * from oras_ro  
union all  
select * from oras_other@bd_other);
```

-- b inclus in a

```
(select * from oras_ro  
union all  
select * from oras_other@bd_other)  
minus  
select *  
from modbd_centralizat.oras;
```

--disjunctia

```
select * from oras_ro  
intersect  
select * from oras_other@bd_other;
```

```

--ORAS
SELECT *
FROM modbd_centralizat.oras;

create table oras_ro
as
select a.*
from modbd_centralizat.oras a
where EXISTS
(select 1
from tara_ro b
where a.id_tara = b.id_tara
);

select * from oras_ro;

```

Script Output x Explain Plan x Query Res

SQL | All Rows Fetched: 14 in 0.003

ID_ORAS	NUME_ORAS	ID_TARA
1	60 Battambang	18
2	406 Phnom Penh	18
3	601 Botosani	18
4	602 Bucuresti	18
5	603 Iasi	18
6	604 Sibiu	18
7	605 Cluj	18
8	606 Brasov	18
9	607 Constanta	18
10	608 Ploiesti	18
11	609 Pitesti	18
12	610 Arad	18
13	611 Oradea	18
14	612 Timisoara	18

```

--HOTEL
SELECT *
FROM modbd_centralizat.hotel;

```

```

create table hotel_ro
as
select a.*
from modbd_centralizat.hotel a
where EXISTS
(select 1
from oras_ro b

```

```

where a.id_oras = b.id_oras
);

select * from hotel_ro;

--completitudine
select *
from modbd_centralizat.hotel
minus
(select * from hotel_ro
union all
select * from hotel_other@bd_other);

--reconstructia
-- a=b
-- a inclus in b
select *
from modbd_centralizat.hotel
minus
(select * from hotel_ro
union all
select * from hotel_other@bd_other);

-- b inclus in a

(select * from hotel_ro
union all
select * from hotel_other@bd_other)
minus
select *
from modbd_centralizat.hotel;

--disjunctia
select * from hotel_ro
intersect
select * from hotel_other@bd_other;

--CAMERA
SELECT *
FROM modbd_centralizat.camera;

create table camera_ro
as
select a.*
from modbd_centralizat.camera a
where EXISTS
(select 1
from hotel_ro b

```

```

where a.id_hotel = b.id_hotel
);

select * from camera_ro;

--CAMERA_REZERVARE
SELECT *
FROM modbd_centralizat.camera_rezervare;

create table camera_rezervare_ro
as
select a.*
from modbd_centralizat.camera_rezervare a
where EXISTS
(select 1
from camera_ro b
where a.id_camera = b.id_camera
);

select * from camera_rezervare_ro;

```

```

105 --HOTEL
106 SELECT *
107 FROM modbd_centralizat.hotel;
108
109 create table hotel_ro
110 as
111 select a.*
112 from modbd_centralizat.hotel a
113 where EXISTS
114 (select 1
115  from oras_ro b
116  where a.id_oras = b.id_oras
117 );
118
119 select * from hotel_ro;
120
121 --CAMERA

```

Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 8 in 0.01 seconds

ID_HOTEL	ID_ORAS	ID_TIP_HOTEL	NUME	NR_STELE	CAPACITATE
1	1	601	4 Kellington	3	300
2	2	602	3 Skeeles	2	420
3	3	603	1 Shields	1	310
4	4	604	5 Vaneschi	4	350
5	5	605	4 McGaraghan	1	330
6	6	606	2 Grelik	3	340
7	7	607	3 Wilshaw	5	390
8	8	608	1 Aps	4	420

```

.21 --CAMERA
.22 SELECT *
.23 FROM modbd_centralizat.camera;
.24
.25 create table camera_ro
.26 as
.27 select a.*
.28 from modbd_centralizat.camera a
.29 where EXISTS
.30 (select 1
.31 from hotel_ro b
.32 where a.id_hotel = b.id_hotel
.33 );
.34
.35 select * from camera_ro;
.36

```

Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 13 in 0.008 seconds

ID_CAMERA	ID_HOTEL	ID_TIP_CAMERA	NUME	PRET	BALCON
1	1	1	2 Apricots - Dried	100	0
2	9	1	2 Lettuce - Mini Greens, Whole	250	1
3	2	2	3 Bread - Hot Dog Buns	330	0
4	10	2	1 Spring Roll Veg Mini	220	1
5	11	2	2 Brandy - Bar	360	1
6	3	3	4 Truffle - Whole Black Peeled	100	1
7	4	4	3 Veal - Slab Bacon	320	0
8	5	5	2 Tuna - Sushi Grade	360	0
9	6	6	4 Wine - Barossa Valley Estate	120	0
10	7	7	3 Guinea Fowl	300	0
11	8	7	2 Pastry - Carrot Muffin - Mini	490	0
12	12	8	4 Vinegar - Red Wine	110	0
13	13	8	2 Eggroll	220	1

```

6
7 --CAMERA_REZERVARE
8 SELECT *
9 FROM modbd_centralizat.camera_rezervare;
10
11 create table camera_rezervare_ro
12 as
13 select a.*
14 from modbd_centralizat.camera_rezervare
15 where EXISTS
16 (select 1
17 from camera_ro b
18 where a.id_camera = b.id_camera
19 );
20
21 select * from camera_rezervare_ro;
22

```

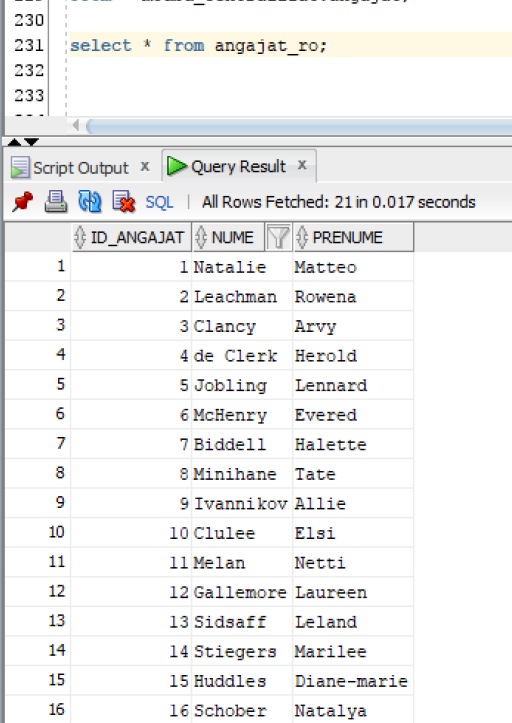
Script Output x Explain Plan x Query Result x

SQL | All Rows Fetched: 168 in 0.009 seconds

ID_REZERVARE	ID_CAMERA	PRET_PER_NOAPTE	
1	25	1	6100
2	61	1	9700
3	73	1	4900
4	86	1	8400
5	95	1	1800
6	101	1	1100
7	122	1	3600
8	133	1	2400
9	183	1	6100
10	184	1	8500
11	196	1	6400
12	209	1	9900
13	210	1	2000
14	221	1	3100
15	232	1	4600

```
-----  
--fragmentare verticala ANGAJAT  
-----
```

```
create table angajat_ro  
as  
select id_angajat, nume, prenume  
from modbd_centralizat.angajat;  
  
select * from angajat_ro;
```



230
231 `select * from angajat_ro;`
232
233

Script Output x Query Result x

All Rows Fetched: 21 in 0.017 seconds

ID_ANGAJAT	NUME	PRENUME
1	1 Natalie	Matteo
2	2 Leachman	Rowena
3	3 Clancy	Arvy
4	4 de Clerk	Herold
5	5 Jobling	Lennard
6	6 McHenry	Evered
7	7 Biddell	Halette
8	8 Minihane	Tate
9	9 Ivannikov	Allie
10	10 Clulee	Elsi
11	11 Melan	Netti
12	12 Gallemore	Laureen
13	13 Sidsaff	Leland
14	14 Stiegers	Marilee
15	15 Huddles	Diane-marie
16	16 Schober	Natalya

```
-----  
--replicare  
-----
```

```
--REZERVARE
```

```
create table rezervare  
as  
select *  
from modbd_centralizat.rezervare;  
  
select * from rezervare;  
  
--PLATA
```

```
create table plata
as
select *
from modbd_centralizat.plata;

select * from plata;

--TIP_HOTEL
create table tip_hotel
as
select *
from modbd_centralizat.tip_hotel;

select * from tip_hotel;

--TIP_CAMERA
create table tip_camera
as
select *
from modbd_centralizat.tip_camera;

select * from tip_camera;

--DISCOUNT
create table discount
as
select *
from modbd_centralizat.discount;

select * from discount;

--CLIENT
create table client
as
select *
from modbd_centralizat.client;

select * from client;
```

263

264 `select * from tip_hotel;`

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.0

ID_TIP_HOTEL	NUME_TIP
1	1 H
2	2 P
3	3 R
4	4 V
5	5 M

2) BD_OTHER

 --fragmentare orizontala primara TARA

```
CREATE TABLE tara_other AS
SELECT *
FROM modbd_centralizat.tara@bd_ro
WHERE nume_tara != 'Romania';
```

```
select * from tara_other;
```

```
select * from tara_other
union all
select * from tara_ro@bd_ro;
```



```

34
35 select * from tara_other;
36
37 select * from tara_other

```

Script Output x Query Result x

SQL | All Rows Fetched: 17 in 0.00

ID_TARA	NUME_TARA
1	1 Afghanistan
2	2 Algeria
3	3 American Samoa
4	4 Angola
5	5 Anguilla
6	6 Argentina
7	7 Armenia
8	8 Australia
9	9 Austria
10	10 Azerbaijan
11	11 Bahrain
12	12 Bangladesh
13	13 Belarus
14	14 Bolivia
15	15 Brazil
16	16 Brunei

```

-----
--fragmentare orizontala derivata
-----

```

```

--ORAS
SELECT *
FROM modbd_centralizat.oras@bd_ro;

```

```

create table oras_other
as
select a.*
from modbd_centralizat.oras@bd_ro a
where EXISTS
(select 1
from tara_other b
where a.id_tara = b.id_tara
);

```

```

select * from oras_other;

```

```

--HOTEL
SELECT *
FROM modbd_centralizat.hotel@bd_ro;

```

```
create table hotel_other
as
select a.*
from modbd_centralizat.hotel@bd_ro a
where EXISTS
(select 1
from oras_other b
where a.id_oras = b.id_oras
);

select * from hotel_other;

--CAMERA
SELECT *
FROM modbd_centralizat.camera@bd_ro;

create table camera_other
as
select a.*
from modbd_centralizat.camera@bd_ro a
where EXISTS
(select 1
from hotel_other b
where a.id_hotel = b.id_hotel
);

select * from camera_other;

--CAMERA_REZERVARE
SELECT *
FROM modbd_centralizat.camera_rezervare@bd_ro;

create table camera_rezervare_other
as
select a.*
from modbd_centralizat.camera_rezervare@bd_ro a
where EXISTS
(select 1
from camera_other b
where a.id_camera = b.id_camera
);

select * from camera_rezervare_other;
```

```

41
42 --ORAS
43 SELECT *
44 FROM modbd_centralizat.oras@bd_ro;
45
46 create table oras_other
47 as
48 select a.*
49 from modbd_centralizat.oras@bd_ro a
50 where EXISTS
51 (select 1
52 from tara_other b
53 where a.id_tara = b.id_tara
54 );
55
56 select * from oras_other;
57
58 --HOTEL
59 SELECT *

```

Script Output x Query Result x		
SQL All Rows Fetched: 67 in 0.015 seconds		
ID_ORAS	NUME_ORAS	ID_TARA
1	251 Kabul	1
2	59 Batna	2
3	63 Bchar	2
4	483 Skikda	2
5	516 Tafuna	3
6	67 Benguela	4
7	360 Namibe	4
8	493 South Hill	5
9	20 Almirante Brown	6
10	43 Avellaneda	6
11	45 Baha Blanca	6
12	128 Crdoba	6
13	161 Escobar	6
14	165 Ezeiza	6
15	289 La Plata	6





```
-----  
--fragmentare verticala  
-----
```

```
create table angajat_other  
as  
select id_angajat, salariu, data_angajare, nr_telefon  
from modbd_centralizat.angajat@bd_ro;  
  
select * from angajat_other;
```

```
111 |  
112 |-----  
113 |--fragmentare verticala  
114 |-----  
115 |  
116 |create table angajat_other  
117 |as  
118 |select id_angajat, salariu, data_angajare, nr_telefon  
119 |from modbd_centralizat.angajat@bd_ro;  
120 |  
121 |select * from angajat_other;  
122 |  
123 |  
124 |-----
```

Script Output x

Query Result x

 SQL | All Rows Fetched: 21 in 0.003 seconds

	ID_ANGAJAT	SALARIU	DATA_ANGAJARE	NR_TELEFON
1	1	1800	02-MAY-98	6782735265
2	2	3700	29-OCT-91	9492566846
3	3	600	31-DEC-95	2858446803
4	4	9500	07-APR-99	3076494005
5	5	1200	25-AUG-91	2981161603
6	6	1200	21-DEC-94	2693830618
7	7	2500	24-JAN-91	4289512482
8	8	8700	25-MAR-93	5071176069
9	9	6400	08-APR-93	5258077722
10	10	5300	31-AUG-99	2801141423
11	11	7400	26-SEP-97	7738888536
12	12	1400	30-APR-95	3018996347
13	13	6500	19-NOV-97	2179960647
14	14	8900	29-AUG-94	3209537109
15	15	4400	08-JUL-90	4904309419
16	16	3000	21-NOV-90	8256874453

--replicare

--REZERVARE

```
create table rezervare
as
select *
from modbd_centralizat.rezervare@bd_ro;

select * from rezervare;
```

--PLATA

```
create table plata
as
select *
from modbd_centralizat.plata@bd_ro;

select * from plata;
```

--TIP_HOTEL

```
create table tip_hotel
as
select *
from modbd_centralizat.tip_hotel@bd_ro;

select * from tip_hotel;
```

--TIP_CAMERA

```
create table tip_camera
as
select *
from modbd_centralizat.tip_camera@bd_ro;

select * from tip_camera;
```

--DISCOUNT

```
create table discount
as
select *
from modbd_centralizat.discount@bd_ro;

select * from discount;
```

--CLIENT

```
create table client
as
select *
from modbd_centralizat.client@bd_ro;

select * from client;
```

3. Furnizarea formelor de transparenta pentru intreg modelul ales

a) Transparenta pentru fragmentele verticale

a. BD_RO

```
--drepturi pentru modbd_global
declare
begin
  FOR x IN (SELECT * FROM user_tables)
  LOOP
    EXECUTE IMMEDIATE 'GRANT ALL ON ' || x.table_name || ' TO modbd_global';
  END LOOP;
end;
/

--a. Transparenta pentru fragmentele verticale
alter table angajat_ro
add primary key (id_angajat);
```

b. BD_OTHER

```
--a. Transparenta pentru fragmentele verticale
alter table angajat_other
add primary key (id_angajat);
```

c. BD_GLOBAL

```
select table_name
from all_tables
where owner = 'MODBD_TRAVEL';

--a. Transparenta pentru fragmentele verticale

--am ales sa stocam datele sensibile ale unui angajat (data_angajare, salariu, nr_telefon) pe statia
bd_other
--iar celelalte date (nume si prenume) pe bd_ro
```

```

create or replace view angajat
as
select a.id_angajat, a.data_angajare, a.salariu, a.nr_telefon,
       b.nume, b.prenume
from   angajat_other@bd_other a, modbd_travel.angajat_ro b
where  a.id_angajat = b.id_angajat;

select * from angajat;

select column_name, updatable, insertable, deletable
from   user_updatable_columns
where  table_name = upper('angajat');

delete from angajat;

rollback;

select * from angajat_other@bd_other;
select * from modbd_travel.angajat_ro;

```

```

CREATE OR REPLACE TRIGGER trigg_angajat
INSTEAD OF INSERT OR DELETE OR UPDATE ON angajat
FOR EACH ROW
BEGIN

    if INSERTING then
        insert into angajat_other@bd_other
        values (:new.id_angajat, :new.salariu, :new.data_angajare, :new.nr_telefon);
        insert into modbd_travel.angajat_ro
        values (:new.id_angajat, :new.nume, :new.prenume);
    elsif DELETING then
        delete from angajat_other@bd_other
        where :old.id_angajat = id_angajat;
        delete from modbd_travel.angajat_ro
        where :old.id_angajat = id_angajat;
    elsif UPDATING then
        update angajat_other@bd_other set salariu = :new.salariu, data_angajare = :new.data_angajare,
        nr_telefon = :new.nr_telefon
        where id_angajat = :OLD.id_angajat;
        update modbd_travel.angajat_ro set nume = :new.nume, prenume = :new.prenume
        where id_angajat = :OLD.id_angajat;
    end if;
END;
/

insert into angajat

```

```
values (999, sysdate, 2000, '0722222222', 'Ion', 'Ion');
```

```
select * from modbd_travel.angajat_ro;
select * from angajat_other@bd_other;
select * from angajat;
rollback;
```

20	20	13-JAN-95	2000	7289149194	Prayer	Shane
21	999	08-MAY-23	2000	0722222222	Ion	Ion

Script Output x Query Result x

SQL | All Rows Fetched: 21 in 0 seconds

ID_ANGAJAT	NUME	PRENUME
9	Ivannikov	Allie
10	Clulee	Elsi
11	Melan	Netti
12	Gallemore	Laureen
13	Sidsaff	Leland
14	Stiegers	Marilee
15	Huddles	Diane-marie
16	Schober	Natalya
17	Estevez	Melisenda
18	Lemme	Jennifer
19	Baldree	Lucie
20	Prayer	Shane
21	999 Ion	Ion

66

Script Output x Query Result x

SQL | All Rows Fetched: 21 in 0.005 seconds

ID_ANGAJAT	SALARIU	DATA_ANGAJARE	NR_TELEFON
9	6400	08-APR-93	5258077722
10	5300	31-AUG-99	2801141423
11	7400	26-SEP-97	7738888536
12	1400	30-APR-95	3018996347
13	6500	19-NOV-97	2179960647
14	8900	29-AUG-94	3209537109
15	4400	08-JUL-90	4904309419
16	3000	21-NOV-90	8256874453
17	5400	26-MAY-93	2347137505
18	6600	20-MAR-99	9407441219
19	6800	05-MAR-92	2108438048
20	2000	13-JAN-95	7289149194
21	999	2000 08-MAY-23	0722222222

b) Transparența pentru fragmentele orizontale

```
--TARA
select * from modbd_travel.tara_ro
union all
select * from modbd_travel.tara_other@bd_other;

create or replace view tara
as
select * from modbd_travel.tara_ro
union all
select * from modbd_travel.tara_other@bd_other;

select * from tara;

select column_name, updatable, insertable, deletable
from user_updatable_columns
where table_name = upper('tara');

delete from tara;

--creare trigger care sa permita propagarea operatiilor insert, delete, update

CREATE OR REPLACE TRIGGER trigg_tara
INSTEAD OF INSERT OR DELETE OR UPDATE ON tara
FOR EACH ROW
BEGIN

    if INSERTING then
        if :new.nume_tara != 'Romania' then
            insert into tara_other@bd_other
            values (:new.id_tara, :new.nume_tara);
        else
            insert into modbd_travel.tara_ro
            values (:new.id_tara, :new.nume_tara);
        end if;
    elsif DELETING then
        if :old.nume_tara != 'Romania' then
            delete from tara_other@bd_other
            where :old.id_tara = id_tara;
        else
            delete from modbd_travel.tara_ro
            where :old.id_tara = id_tara;
        end if;
    elsif UPDATING then
        if :old.nume_tara != 'Romania' then
```

```
        update tara_other@bd_other set nume_tara = :new.nume_tara
        where id_tara = :OLD.id_tara;
    else
        update modbd_travel.tara_ro set nume_tara = :new.nume_tara
        where id_tara = :OLD.id_tara;
    end if;
end if;
END;
/
```

```
insert into tara
values(999, 'Africa de sud');
```

```
select * from tara_other@bd_other;
select * from tara;
```

```
rollback;
```

```
delete from tara
where nume_tara = 'Argentina';
```

```
select * from tara_other@bd_other;
select * from tara;
```

```
rollback;
```

```
update tara
set nume_tara = 'Argentina 2'
where nume_tara = 'Argentina';
```

```
select * from tara_other@bd_other;
select * from tara;
```

```
rollback;
```

```

101 select * from tara;
102
103 rollback;

```

Script Output x Query Result x

SQL | All Rows Fetched: 18 in 0.025 sec

ID_TARA	NUME_TARA
1	18 Romania
2	1 Afghanistan
3	2 Algeria
4	3 American Samoa
5	4 Angola
6	5 Anguilla
7	6 Argentina 2
8	7 Armenia

```

95
96 update tara
97 set nume_tara = 'Argentina 2'
98 where nume_tara = 'Argentina';
99
100 select * from tara_other@bd_other;
101 select * from tara;
102
103 rollback;

```

Script Output x Query Result x

SQL | All Rows Fetched: 17 in 0.006 seconds

ID_TARA	NUME_TARA
1	1 Afghanistan
2	2 Algeria
3	3 American Samoa
4	4 Angola
5	5 Anguilla
6	6 Argentina 2
7	7 Armenia
8	8 Australia

--ORAS

```

create or replace view oras
as
select * from modbd_travel.oras_ro
union all
select * from modbd_travel.oras_other@bd_other;

select * from oras;

```

```

select column_name, updatable, insertable, deletable
from user_updatable_columns
where table_name = upper('oras');

--creare trigger care sa permita propagarea operatiilor insert, delete, update

CREATE OR REPLACE TRIGGER trigg_oras
INSTEAD OF INSERT OR DELETE OR UPDATE ON oras
FOR EACH ROW
DECLARE
    v_nume_tara varchar2(50);
BEGIN

    if INSERTING then
        select t.nume_tara into v_nume_tara
        from tara t
        where t.id_tara = :new.id_tara;

        if v_nume_tara != 'Romania' then
            insert into oras_other@bd_other
            values (:new.id_oras, :new.nume_oras, :new.id_tara);
        else
            insert into modbd_travel.oras_ro
            values (:new.id_oras, :new.nume_oras, :new.id_tara);
        end if;
    elsif DELETING then
        select t.nume_tara into v_nume_tara
        from tara t
        where t.id_tara = :old.id_tara;
        if v_nume_tara != 'Romania' then
            delete from oras_other@bd_other
            where :old.id_oras = id_oras;
        else
            delete from modbd_travel.oras_ro
            where :old.id_oras = id_oras;
        end if;
    elsif UPDATING then
        select t.nume_tara into v_nume_tara
        from tara t
        where t.id_tara = :new.id_tara;
        if v_nume_tara != 'Romania' then
            update oras_other@bd_other set nume_oras = :new.nume_oras
            where id_oras = :OLD.id_oras;
        else
            update modbd_travel.oras_ro set nume_oras = :new.nume_oras
            where id_oras = :OLD.id_oras;
        end if;
    end if;
END;

```

```

        end if;
    end if;
END;
/

select * from oras;

insert into oras
values(999, 'Oras 1', 18);

select * from modbd_travel.oras_ro;
select * from oras;

rollback;

```

```

:18 |
:19 | select * from modbd_travel.oras_ro;
:20 | select * from oras;
:21 |

```

Script Output x Query Result x			
SQL All Rows Fetched: 15 in 0.003 seconds			
	ID_ORAS	NUME_ORAS	ID_TARA
1	600	Timisoara	18
2	601	Botosani	18
3	601	Botosani	18
4	602	Bucuresti	18
5	603	Iasi	18
6	604	Sibiu	18
7	605	Cluj	18
8	606	Brasov	18
9	607	Constanta	18
10	608	Ploiesti	18
11	609	Pitesti	18
12	610	Arad	18
13	611	Oradea	18
14	612	Timisoara	18
15	999	Oras 1	18

```

219 | select * from modbd_travel.oras_ro;
220 | select * from oras where id_oras = 999;
221 |
222 | rollback;

```

Script Output x Query Result x			
SQL All Rows Fetched: 1 in 0.009 seconds			
	ID_ORAS	NUME_ORAS	ID_TARA
1	999	Oras 1	18

--HOTEL

create or replace view hotel

as

select * from modbd_travel.hotel_ro

union all

select * from modbd_travel.hotel_other@bd_other;

select * from hotel;

select column_name, updatable, insertable, deletable

from user_updatable_columns

where table_name = upper('hotel');

--create trigger care sa permita propagarea operatiilor insert, delete, update

CREATE OR REPLACE TRIGGER trigg_hotel

INSTEAD OF INSERT OR DELETE OR UPDATE ON hotel

FOR EACH ROW

DECLARE

 v_nume_tara varchar2(50);

BEGIN

 if INSERTING then

 select t.nume_tara into v_nume_tara

 from tara t

 join oras o on(t.id_tara = o.id_tara)

 where o.id_oras = :new.id_oras;

 if v_nume_tara != 'Romania' then

 insert into hotel_other@bd_other

 values (:new.id_hotel, :new.id_oras, :new.id_tip_hotel, :new.nume, :new.nr_stele,
:new.capacitate);

 else

 insert into modbd_travel.hotel_ro

 values (:new.id_hotel, :new.id_oras, :new.id_tip_hotel, :new.nume, :new.nr_stele,
:new.capacitate);

 end if;

 elsif DELETING then

 select t.nume_tara into v_nume_tara

 from tara t

 join oras o on(t.id_tara = o.id_tara)

 join hotel h on (h.id_oras = o.id_oras)

 where h.id_hotel = :old.id_hotel;

```

if v_nume_tara != 'Romania' then
    delete from hotel_other@bd_other
    where :old.id_hotel = id_hotel;
else
    delete from modbd_travel.hotel_ro
    where :old.id_hotel = id_hotel;
end if;
elsif UPDATING then
    select t.nume_tara into v_nume_tara
    from tara t
    join oras o on (t.id_tara = o.id_tara)
    join hotel h on (h.id_oras = o.id_oras)
    where h.id_hotel = :old.id_hotel;

    if v_nume_tara != 'Romania' then
        update hotel_other@bd_other set nr_stele = :new.nr_stele, capacitate = :new.capacitate
        where id_hotel = :OLD.id_hotel;
    else
        update modbd_travel.hotel_ro set nr_stele = :new.nr_stele, capacitate = :new.capacitate
        where id_hotel = :OLD.id_hotel;
    end if;
end if;
END;
/

select * from hotel;

insert into hotel
values(999, 602, 1, 'Nume hotel', 3, 200);

select * from modbd_travel.hotel_ro;
select * from hotel;

rollback;

```

--CAMERA

```

create or replace view camera
as
select * from modbd_travel.camera_ro
union all
select * from modbd_travel.camera_other@bd_other;

select * from camera;

select column_name, updatable, insertable, deletable

```

```
from user_updatable_columns
where table_name = upper('camera');
```

--create trigger care sa permita propagarea operatiilor insert, delete, update

```
CREATE OR REPLACE TRIGGER trigg_camera
INSTEAD OF INSERT OR DELETE OR UPDATE ON camera
FOR EACH ROW
DECLARE
    v_nume_tara varchar2(50);
BEGIN
```

```
    if INSERTING then
        select t.nume_tara into v_nume_tara
        from tara t
        join oras o on(t.id_tara = o.id_tara)
        join hotel h on(o.id_oras = h.id_oras)
        where h.id_hotel = :new.id_hotel;

        if v_nume_tara != 'Romania' then
            insert into camera_other@bd_other
            values (:new.id_camera, :new.id_hotel, :new.id_tip_camera, :new.nume, :new.pret,
: new.balcon);
        else
            insert into modbd_travel.camera_ro
            values (:new.id_camera, :new.id_hotel, :new.id_tip_camera, :new.nume, :new.pret,
: new.balcon);
        end if;
    elsif DELETING then
        select t.nume_tara into v_nume_tara
        from tara t
        join oras o on(t.id_tara = o.id_tara)
        join hotel h on(o.id_oras = h.id_oras)
        join camera c on(c.id_hotel = h.id_hotel)
        where c.id_camera = :old.id_camera;

        if v_nume_tara != 'Romania' then
            delete from camera_other@bd_other
            where :old.id_camera = id_camera;
        else
            delete from modbd_travel.camera_ro
            where :old.id_camera = id_camera;
        end if;
    elsif UPDATING then
        select t.nume_tara into v_nume_tara
        from tara t
```



```

join oras o on(t.id_tara = o.id_tara)
join hotel h on(o.id_oras = h.id_oras)
join camera c on(c.id_hotel = h.id_hotel)
where c.id_camera = :new.id_camera;

if v_nume_tara != 'Romania' then
    update camera_other@bd_other set pret = :new.pret, balcon = :new.balcon
    where id_camera = :OLD.id_camera;
else
    update modbd_travel.camera_ro set pret = :new.pret, balcon = :new.balcon
    where id_camera = :OLD.id_camera;
end if;
end if;
END;
/

select * from camera;

insert into camera
values(999, 9, 1, 'Nume camera', 300, 1);

select * from camera_other@bd_other;
select * from camera;

rollback;

```

```

--CAMERA_REZERVARE

create or replace view camera_rezervare
as
select * from modbd_travel.camera_rezervare_ro
union all
select * from modbd_travel.camera_rezervare_other@bd_other;

select * from camera_rezervare;

select column_name, updatable, insertable, deletable
from user_updatable_columns
where table_name = upper('camera_rezervare');

--create trigger care sa permita propagarea operatiilor insert, delete, update

CREATE OR REPLACE TRIGGER trigg_camera_rezervare
INSTEAD OF INSERT OR DELETE OR UPDATE ON camera_rezervare
FOR EACH ROW
DECLARE

```

```

v_nume_tara varchar2(50);
BEGIN

if INSERTING then
    select t.nume_tara into v_nume_tara
    from tara t
    join oras o on(t.id_tara = o.id_tara)
    join hotel h on(o.id_oras = h.id_oras)
    join camera c on(c.id_hotel = h.id_hotel)
    where c.id_camera = :new.id_camera;

    if v_nume_tara != 'Romania' then
        insert into camera_rezervare_other@bd_other
        values (:new.id_rezervare, :new.id_camera, :new.pret_per_noapte);
    else
        insert into modbd_travel.camera_rezervare_ro
        values (:new.id_rezervare, :new.id_camera, :new.pret_per_noapte);
    end if;
elsif DELETING then
    select t.nume_tara into v_nume_tara
    from tara t
    join oras o on(t.id_tara = o.id_tara)
    join hotel h on(o.id_oras = h.id_oras)
    join camera c on(c.id_hotel = h.id_hotel)
    join camera_rezervare cr on(cr.id_camera = c.id_camera)
    where cr.id_camera = :old.id_camera;

    if v_nume_tara != 'Romania' then
        delete from camera_rezervare_other@bd_other
        where :old.id_camera = id_camera;
    else
        delete from modbd_travel.camera_rezervare_ro
        where :old.id_camera = id_camera;
    end if;
elsif UPDATING then
    select t.nume_tara into v_nume_tara
    from tara t
    join oras o on(t.id_tara = o.id_tara)
    join hotel h on(o.id_oras = h.id_oras)
    join camera c on(c.id_hotel = h.id_hotel)
    join camera_rezervare cr on(cr.id_camera = c.id_camera)
    where cr.id_camera = :old.id_camera;

    if v_nume_tara != 'Romania' then
        update camera_rezervare_other@bd_other set pret_per_noapte = :new.pret_per_noapte
        where id_camera = :OLD.id_camera;
    else
        update modbd_travel.camera_rezervare_ro set pret_per_noapte = :new.pret_per_noapte

```

```

        where id_camera = :OLD.id_camera;
    end if;
end if;
END;
/

select * from camera_rezervare;

insert into camera_rezervare
values(999, 9, 1000);

select * from modbd_travel.camera_rezervare_ro;
select * from camera_rezervare
order by 1;

rollback;

```

c) Transparența pentru fragmentele replicate

```

--TIP_HOTEL

create or replace view tip_hotel
as
select * from modbd_travel.tip_hotel;

create or replace trigger trigg_rep_tip_hotel
instead of insert or update or delete on tip_hotel
for each row
begin
    if inserting then
        insert into modbd_travel.tip_hotel
        values(:new.id_tip_hotel, :new.nume_tip);
    ELSif deleting then
        delete from modbd_travel.tip_hotel
        where id_tip_hotel = :old.id_tip_hotel;
    else
        update modbd_travel.tip_hotel
        set nume_tip = :new.nume_tip
        where id_tip_hotel = :old.id_tip_hotel;
    end if;
end;
/

--TIP_CAMERA

```

```
create or replace view tip_camera
as
select * from modbd_travel.tip_camera;
```

```
create or replace trigger trigg_rep_tip_camera
instead of insert or update or delete on tip_camera
for each row
begin
    if inserting then
        insert into modbd_travel.tip_camera
        values(:new.id_tip_camera, :new.nume_tip_camera, :new.capacitate);
    ELSif deleting then
        delete from modbd_travel.tip_camera
        where id_tip_camera = :old.id_tip_camera;
    else
        update modbd_travel.tip_camera
        set nume_tip_camera = :new.nume_tip_camera,
            capacitate = :new.capacitate
        where id_tip_camera = :old.id_tip_camera;
    end if;
end;
/
```

--CLIENT

```
create or replace view client
as
select * from modbd_travel.client;
```

```
create or replace trigger trigg_rep_client
instead of insert or update or delete on client
for each row
begin
    if inserting then
        insert into modbd_travel.client
        values(:new.id_client, :new.nume, :new.prenume, :new.data_nastere, :new.nr_telefon,
:new.email);
    ELSif deleting then
        delete from modbd_travel.client
        where id_client = :old.id_client;
    else
        update modbd_travel.client
        set nr_telefon = :new.nr_telefon,
            email = :new.nr_telefon,
            nume = :new.nume
        where id_client = :old.id_client;
```

```

    end if;
end;
/

--REZERVARE

create or replace view rezervare
as
select * from modbd_travel.rezervare;

create or replace trigger trigg_rep_rezervare
instead of insert or update or delete on rezervare
for each row
begin
    if inserting then
        insert into modbd_travel.rezervare
        values(:new.id_rezervare, :new.id_client, :new.id_discount, :new.id_angajat, :new.data_rezervare,
        :new.data_checkin,
            :new.data_checkout, :new.nr_nopti, :new.nr_camere, :new.pret_total, :new.status);
    ELSif deleting then
        delete from modbd_travel.rezervare
        where id_rezervare = :old.id_rezervare;
    else
        update modbd_travel.rezervare
        set data_checkin = :new.data_checkin,
            data_checkout = :new.data_checkout,
            nr_camere = :new.nr_camere,
            nr_nopti = :new.nr_nopti,
            pret_total = :new.pret_total,
            status = :new.status
        where id_rezervare = :old.id_rezervare;
    end if;
end;
/

--PLATA

create or replace view plata
as
select * from modbd_travel.plata;

create or replace trigger trigg_rep_plata
instead of insert or update or delete on plata
for each row
begin
    if inserting then
        insert into modbd_travel.plata

```

```

values(:new.id_plata, :new.id_rezervare, :new.suma, :new.tip_plata, :new.data);
ELSif deleting then
    delete from modbd_travel.plata
    where id_plata = :old.id_plata;
else
    update modbd_travel.plata
    set tip_plata = :new.tip_plata
    where id_plata = :old.id_plata;
end if;
end;
/

--DISCOUNT

create or replace view discount
as
select * from modbd_travel.discount;

create or replace trigger trigg_rep_discount
instead of insert or update or delete on discount
for each row
begin
    if inserting then
        insert into modbd_travel.discount
        values(:new.id_discount, :new.denumire, :new.procent, :new.data_start, :new.data_final);
    ELSif deleting then
        delete from modbd_travel.discount
        where id_discount = :old.id_discount;
    else
        update modbd_travel.discount
        set denumire = :new.denumire,
            procent = :new.procent,
            data_start = :new.data_start,
            data_final = :new.data_final
        where id_discount = :old.id_discount;
    end if;
end;
/

insert into discount
values(99, 'discount 99', 20, sysdate, sysdate + 10);

select * from discount;
select * from discount@bd_other;
select * from modbd_travel.discount;

delete from discount

```

where id_discount = 99;

```
15
16 insert into discount
17 values(99, 'discount 99', 20, sysdate, sysdate + 10);
18
19 select * from discount;
20 select * from discount@bd other;
21 select * from modbd_travel.discount;
22
23 delete from discount
24 where id_discount = 1000;
25
```

ID_DISCOUNT	DENUMIRE	PROCENT	DATA_START	DATA_FINAL
1	1 Discount Iunie 2020	30	01-JUN-20	01-JUL-20
2	2 Discount Iulie 2022	50	01-JUL-22	01-AUG-22
3	3 Discount Ianuarie 2022	40	01-JAN-20	01-FEB-20
4	4 Discount Aprilie 2020	50	01-APR-20	01-MAY-20
5	5 Discount Decembrie 2020	40	01-DEC-20	01-JAN-21
6	6 Discount Aprilie 2022	10	01-APR-22	01-MAY-22
7	7 Discount Martie 2021	50	01-MAR-21	01-APR-21
8	8 Discount Iunie 2021	10	01-JUN-21	01-JUL-21
9	9 Discount Ianuarie 2022	20	01-JAN-22	01-FEB-22
10	10 Discount Noiembrie 2021	10	01-NOV-21	01-DEC-21
11	99 discount 99	20	09-MAY-23	19-MAY-23

```
26 select * from discount;
27 select * from discount@bd other;
28 select * from modbd_travel.discount;
```

ID_DISCOUNT	DENUMIRE	PROCENT	DATA_START	DATA_FINAL
1	1 Discount Iunie 2020	30	01-JUN-20	01-JUL-20
2	2 Discount Iulie 2022	50	01-JUL-22	01-AUG-22
3	3 Discount Ianuarie 2022	40	01-JAN-20	01-FEB-20
4	4 Discount Aprilie 2020	50	01-APR-20	01-MAY-20
5	5 Discount Decembrie 2020	40	01-DEC-20	01-JAN-21
6	6 Discount Aprilie 2022	10	01-APR-22	01-MAY-22
7	7 Discount Martie 2021	50	01-MAR-21	01-APR-21
8	8 Discount Iunie 2021	10	01-JUN-21	01-JUL-21
9	9 Discount Ianuarie 2022	20	01-JAN-22	01-FEB-22
10	10 Discount Noiembrie 2021	10	01-NOV-21	01-DEC-21
11	99 discount 99	20	09-MAY-23	19-MAY-23

```

640 select * from discount@bd_other;
641 select * from modbd_travel.discount;
642
643 delete from discount

```

ID_DISCOUNT	DENUMIRE	PROCENT	DATA_START	DATA_FINAL
1	1 Discount Iunie 2020	30	01-JUN-20	01-JUL-20
2	2 Discount Iulie 2022	50	01-JUL-22	01-AUG-22
3	3 Discount Ianuarie 2022	40	01-JAN-20	01-FEB-20
4	4 Discount Aprilie 2020	50	01-APR-20	01-MAY-20
5	5 Discount Decembrie 2020	40	01-DEC-20	01-JAN-21
6	6 Discount Aprilie 2022	10	01-APR-22	01-MAY-22
7	7 Discount Martie 2021	50	01-MAR-21	01-APR-21
8	8 Discount Iunie 2021	10	01-JUN-21	01-JUL-21
9	9 Discount Ianuarie 2022	20	01-JAN-22	01-FEB-22
10	10 Discount Noiembrie 2021	10	01-NOV-21	01-DEC-21
11	99 discount 99	20	09-MAY-23	19-MAY-23

4. Asigurarea sincronizarii datelor pentru relatiile replicate

```

create or replace trigger trigg_rep_discount
after insert or update or delete on discount
for each row
begin
    if inserting then
        insert into discount@bd_other
        values(:new.id_discount, :new.denumire, :new.procent, :new.data_start, :new.data_final);
    ELSif deleting then
        delete from discount@bd_other
        where id_discount = :old.id_discount;
    else
        update discount@bd_other
        set denumire = :new.denumire,
            procent = :new.procent,
            data_start = :new.data_start,
            data_final = :new.data_final
        where id_discount = :old.id_discount;
    end if;
end;
/

insert into discount
values(99, 'discount 99', 20, sysdate, sysdate + 10);

```



```

select * from discount;

select * from discount@bd_other;

delete from discount
where id_discount = 99;

-- Vom defini trigger-ul trig_rep_hotel care va determina inserarea informatiilor in tabela tip_hotel din
bd_other odata
-- cu inserarea acestora in tabela tip_hotel din bd_ro

create or replace trigger trigg_rep_hotel
after insert or update or delete on tip_hotel
for each row
begin
    if inserting then
        insert into tip_hotel@bd_other
        values(:new.id_tip_hotel, :new.nume_tip);
    ELSif deleting then
        delete from tip_hotel@bd_other
        where id_tip_hotel = :old.id_tip_hotel;
    else
        update tip_hotel@bd_other
        set nume_tip = :new.nume_tip
        where id_tip_hotel = :old.id_tip_hotel;
    end if;
end;
/

-- Vom defini trigger-ul trig_rep_camera care va determina inserarea informatiilor in tabela tip_camera
din bd_other odata
-- cu inserarea acestora in tabela tip_camera din bd_ro

create or replace trigger trigg_rep_camera
after insert or update or delete on tip_camera
for each row
begin
    if inserting then
        insert into tip_camera@bd_other
        values(:new.id_tip_camera, :new.nume_tip_camera, :new.capacitate);
    ELSif deleting then
        delete from tip_camera@bd_other
        where id_tip_camera = :old.id_tip_camera;
    else
        update tip_camera@bd_other
        set nume_tip_camera = :new.nume_tip_camera,
        capacitate = :new.capacitate

```

```

        where id_tip_camera = :old.id_tip_camera;
    end if;
end;
/

-- Vom defini trigger-ul trig_rep_client care va determina inserarea informatiilor in tabela client din
bd_other odata
-- cu inserarea acestora in tabela client din bd_ro

create or replace trigger trig_rep_client
after insert or update or delete on client
for each row
begin
    if inserting then
        insert into client@bd_other
        values(:new.id_client, :new.num, :new.prenume, :new.data_nastere, :new.nr_telefon,
:new.email);
    ELSif deleting then
        delete from client@bd_other
        where id_client = :old.id_client;
    else
        update client@bd_other
        set nr_telefon = :new.nr_telefon,
            email = :new.nr_telefon,
            num = :new.num
        where id_client = :old.id_client;
    end if;
end;
/

-- Vom defini trigger-ul trigg_rep_plata care va determina inserarea informatiilor in tabela plata din
bd_other odata
-- cu inserarea acestora in tabela plata din bd_ro

create or replace trigger trigg_rep_plata
after insert or update or delete on plata
for each row
begin
    if inserting then
        insert into plata@bd_other
        values(:new.id_plata, :new.id_rezervare, :new.suma, :new.tip_plata, :new.data);
    ELSif deleting then
        delete from plata@bd_other
        where id_plata = :old.id_plata;
    else
        update plata@bd_other

```

```

        set tip_plata = :new.tip_plata
        where id_plata = :old.id_plata;
    end if;
end;
/

-- Vom defini trigger-ul trig_rep_rezervare care va determina inserarea informatiilor in tabela
rezervare din bd_other odata
-- cu inserarea acestora in tabela rezervare din bd_ro

create or replace trigger trig_rep_rezervare
after insert or update or delete on rezervare
for each row
begin
    if inserting then
        insert into rezervare@bd_other
        values(:new.id_rezervare, :new.id_client, :new.id_discount, :new.id_angajat, :new.data_rezervare,
: new.data_checkin,
            :new.data_checkout, :new.nr_nopti, :new.nr_camere, :new.pret_total, :new.status);
    ELSif deleting then
        delete from rezervare@bd_other
        where id_rezervare = :old.id_rezervare;
    else
        update rezervare@bd_other
        set data_checkin = :new.data_checkin,
            data_checkout = :new.data_checkout,
            nr_camere = :new.nr_camere,
            nr_nopti = :new.nr_nopti,
            pret_total = :new.pret_total,
            status = :new.status
        where id_rezervare = :old.id_rezervare;
    end if;
end;
/

```

```

334
335 insert into discount
336 values(99, 'discount 99', 20, sysdate, sysdate + 10);
337
338 select * from discount;
339
340 select * from discount@bd other;
341

```

Script Output x Query Result x

SQL | All Rows Fetched: 11 in 0.001 seconds

ID_DISCOUNT	DENUMIRE	PROCENT	DATA_START	DATA_FINAL
1	1 Discount Iunie 2020	30	01-JUN-20	01-JUL-20
2	2 Discount Iulie 2022	50	01-JUL-22	01-AUG-22
3	3 Discount Ianuarie 2022	40	01-JAN-20	01-FEB-20
4	4 Discount Aprilie 2020	50	01-APR-20	01-MAY-20
5	5 Discount Decembrie 2020	40	01-DEC-20	01-JAN-21
6	6 Discount Aprilie 2022	10	01-APR-22	01-MAY-22
7	7 Discount Martie 2021	50	01-MAR-21	01-APR-21
8	8 Discount Iunie 2021	10	01-JUN-21	01-JUL-21
9	9 Discount Ianuarie 2022	20	01-JAN-22	01-FEB-22
10	10 Discount Noiembrie 2021	10	01-NOV-21	01-DEC-21
11	99 discount 99	20	09-MAY-23	19-MAY-23

```

339
340 select * from discount@bd other;
341
342 delete from discount

```

Script Output x Query Result x

SQL | All Rows Fetched: 11 in 0.002 seconds

ID_DISCOUNT	DENUMIRE	PROCENT	DATA_START	DATA_FINAL
1	1 Discount Iunie 2020	30	01-JUN-20	01-JUL-20
2	2 Discount Iulie 2022	50	01-JUL-22	01-AUG-22
3	3 Discount Ianuarie 2022	40	01-JAN-20	01-FEB-20
4	4 Discount Aprilie 2020	50	01-APR-20	01-MAY-20
5	5 Discount Decembrie 2020	40	01-DEC-20	01-JAN-21
6	6 Discount Aprilie 2022	10	01-APR-22	01-MAY-22
7	7 Discount Martie 2021	50	01-MAR-21	01-APR-21
8	8 Discount Iunie 2021	10	01-JUN-21	01-JUL-21
9	9 Discount Ianuarie 2022	20	01-JAN-22	01-FEB-22
10	10 Discount Noiembrie 2021	10	01-NOV-21	01-DEC-21
11	99 discount 99	20	09-MAY-23	19-MAY-23

5. Asigurarea tuturor constrangerilor de integritate

a) BD_RO

```
--6. Constrangeri
--Constrangeri de unicitate
--i. unicitate locală
--La nivel local, vom asigura unicitatea prin adaugarea constrangerilor de unicitate pentru

ALTER TABLE oras_ro
ADD CONSTRAINT unique_oras_ro
UNIQUE (nume_oras);

--ii. unicitate globală pe fragmente orizontale

--Trigg_oras_ro prin care se verifica inainte de a insera o noua inregistrare in tabela oras_ro daca
exista o inregistrare avand aceleasi valori pentru nume_oras in tabela distanta oras_other;

create or replace trigger trig_oras_ro
before insert or update of nume_oras on oras_ro
for each row
declare
    nr number(1);
begin
    select count(*) into nr
    from oras_other@bd_other
    where nume_oras = :new.nume_oras;

    if (nr<>0) then
        raise_application_error (-20001,'Constangere de unicitate pe nume_oras
        incalcata. Fragmentul de pe bd_other contine aceeasi valaore');
    end if;
end;
/

--Se vor verifica cei doi triggeri prin inserarea in tabelele oras_ro, respectiv oras_other.

select * from oras_ro
where nume_oras in ('Bucuresti', 'Tafuna');

select * from oras_ro;

insert into oras_ro
values(999, 'Bucuresti', 18);
```

```
insert into oras_ro  
values(999, 'Tafuna', 18);
```

--Constrangeri de cheie primară (la nivel local/global)
--La nivel local

```
alter table tara_ro  
add constraint pk_tara_ro primary key (id_tara);
```

```
alter table oras_ro  
add constraint pk_oras_ro primary key (id_oras);
```

```
alter table hotel_ro  
add constraint pk_hotel_ro primary key (id_hotel);
```

```
alter table camera_ro  
add constraint pk_camera_ro primary key (id_camera);
```

```
alter table camera_rezervare_ro  
add constraint pk_camera_rezervare_ro primary key (id_camera, id_rezervare);
```

```
alter table angajat_ro  
add constraint pk_angajat_ro primary key (id_angajat);
```

--La nivel global
--Se vor asigura constrangerile de cheie primara pentru relatiile tara, oras, hotel, camera,
camera_rezervare si angajat (fragmentate orizontal si vertical) prin utilizarea secventelor.

```
CREATE SEQUENCE sec_tara_ro  
INCREMENT BY 2  
START WITH 7000  
NOCYCLE;
```

```
CREATE SEQUENCE sec_oras_ro  
INCREMENT BY 2  
START WITH 10000  
NOCYCLE;
```

```
CREATE SEQUENCE sec_hotel_ro  
INCREMENT BY 2  
START WITH 1000  
NOCYCLE;
```

```
CREATE SEQUENCE sec_camera_ro  
INCREMENT BY 2  
START WITH 1000  
NOCYCLE;
```

```
CREATE SEQUENCE sec_angajat_ro
INCREMENT BY 2
START WITH 1000
NOCYCLE;

--Constrangeri de chei externe la nivel local
ALTER TABLE oras_ro ADD CONSTRAINT fk_tara_oras_ro
FOREIGN KEY (id_tara)
REFERENCES tara_ro(id_tara);

ALTER TABLE hotel_ro ADD CONSTRAINT fk_oras_hotel_ro
FOREIGN KEY (id_oras)
REFERENCES oras_ro(id_oras);

ALTER TABLE camera_ro ADD CONSTRAINT fk_hotel_camera_ro
FOREIGN KEY (id_hotel)
REFERENCES hotel_ro(id_hotel);

--Constrangeri de validare

ALTER TABLE hotel_ro
ADD CONSTRAINT nr_stele_check_ro
CHECK (nr_stele IN (1,2,3,4,5));

ALTER TABLE hotel_ro
ADD CONSTRAINT capacitate_hotel_check_ro
CHECK (capacitate > 0);

ALTER TABLE camera_ro
ADD CONSTRAINT pret_camera_check_ro
CHECK (pret > 0);

ALTER TABLE camera_ro
ADD CONSTRAINT balcon_camera_check_ro
CHECK (balcon IN (0,1));

ALTER TABLE angajat_ro
ADD CONSTRAINT nume_angajat_check_ro
CHECK (nume NOT LIKE '%[0-9]%');

ALTER TABLE angajat_ro
ADD CONSTRAINT prenume_angajat_check_ro
CHECK (prenume NOT LIKE '%[0-9]%');
```

```

485
486 insert into oras_ro
487 values(999, 'Bucuresti', 18);
488
489 insert into oras_ro
490 values(999, 'Tafuna', 18);
491

```

Script Output x Query Result x

Task completed in 0.025 seconds

values(999, 'Bucuresti', 18)

Error report -

DRA-00001: unique constraint (MODBD_TRAVEL.UNIQUE_ORAS_RO) violated

Error starting at line : 489 in command -

```

insert into oras_ro
values(999, 'Tafuna', 18)
Error report -
DRA-20001: Constangere de unicitate pe nume_oras
          incalcata. Fragmentul de pe bd_other contine aceeasi valoare
DRA-06512: at "MODBD_TRAVEL.TRIG_ORAS_RO", line 9
DRA-04088: error during execution of trigger 'MODBD_TRAVEL.TRIG_ORAS_RO'

```

b) BD_OTHER

```

--6. Constrangeri
--Constrangeri de unicitate
--i. de unicitate locala
ALTER TABLE oras_other
ADD CONSTRAINT unique_oras_other
UNIQUE (nume_oras);

ALTER TABLE angajat_other
ADD CONSTRAINT unique_angajat_other
UNIQUE (nr_telefon);

--ii. unicitate globală pe fragmente orizontale
--Trigg_oras_other prin care se verifica inainte de a insera o noua inregistrare in tabela oras_other
daca exista o inregistrare avand aceleasi valori pentru nume_oras in tabela distanta oras_ro

create or replace trigger trig_oras_other
before insert or update of nume_oras on oras_other
for each row
declare

```



```

    nr number(1);
begin
    select count(*) into nr
    from oras_ro@bd_ro
    where nume_oras = :new.nume_oras;

    if (nr<>0) then
        raise_application_error (-20001,'Constangere de unicitate pe nume_oras
        incalcata. Fragmentul de pe bd_ro contine aceeasi valoare');
    end if;
end;
/

--Se vor verifica cei doi triggeri prin inserarea in tabelele oras_ro, respectiv oras_other.

select * from oras_other;

select * from oras_other
where nume_oras in ('Bucuresti', 'Tafuna');

insert into oras_other
values(999, 'Bucuresti', 2);

insert into oras_other
values(999, 'Tafuna', 2);

--Constrangeri de cheie primară (la nivel local/global)
--La nivel local

alter table tara_other
add constraint pk_tara_other primary key (id_tara);

alter table oras_other
add constraint pk_oras_other primary key (id_oras);

alter table hotel_other
add constraint pk_hotel_other primary key (id_hotel);

alter table camera_other
add constraint pk_camera_other primary key (id_camera);

alter table camera_rezervare_other
add constraint pk_camera_rezervare_other primary key (id_camera, id_rezervare);

alter table angajat_other
add constraint pk_angajat_other primary key (id_angajat);

--La nivel global

```

--Se vor asigura constrangerile de cheie primara pentru relatiile tara, oras, hotel, camera, camera_rezervare si angajat (fragmentate orizontal si vertical) prin utilizarea secventelor.

```
CREATE SEQUENCE sec_tara_other  
INCREMENT BY 2  
START WITH 7001  
NOCYCLE;
```

```
CREATE SEQUENCE sec_oras_other  
INCREMENT BY 2  
START WITH 10001  
NOCYCLE;
```

```
CREATE SEQUENCE sec_hotel_other  
INCREMENT BY 2  
START WITH 1001  
NOCYCLE;
```

```
CREATE SEQUENCE sec_camera_other  
INCREMENT BY 2  
START WITH 1001  
NOCYCLE;
```

```
CREATE SEQUENCE sec_angajat_other  
INCREMENT BY 2  
START WITH 1001  
NOCYCLE;
```

--Constrangeri de chei externe la nivel local

```
ALTER TABLE oras_other ADD CONSTRAINT fk_tara_oras_other  
FOREIGN KEY (id_tara)  
REFERENCES tara_other(id_tara);
```

```
ALTER TABLE hotel_other ADD CONSTRAINT fk_oras_hotel_other  
FOREIGN KEY (id_oras)  
REFERENCES oras_other(id_oras);
```

```
ALTER TABLE camera_other ADD CONSTRAINT fk_hotel_camera_other  
FOREIGN KEY (id_hotel)  
REFERENCES hotel_other(id_hotel);
```

--Constrangeri de validare

```
ALTER TABLE hotel_other  
ADD CONSTRAINT nr_stele_check_other  
CHECK (nr_stele IN (1,2,3,4,5));
```

```
ALTER TABLE hotel_other
```

```
ADD CONSTRAINT capacitate_hotel_check_other  
CHECK (capacitate > 0);
```

```
ALTER TABLE camera_other  
ADD CONSTRAINT pret_camera_check_other  
CHECK (pret > 0);
```

```
ALTER TABLE camera_other  
ADD CONSTRAINT balcon_camera_check_other  
CHECK (balcon IN (0,1));
```

```
CREATE OR REPLACE TRIGGER trg_check_date_other  
BEFORE INSERT OR UPDATE ON angajat_other  
FOR EACH ROW  
BEGIN  
IF( :new.data_angajare >= SYSDATE )  
THEN  
RAISE_APPLICATION_ERROR( -20001,  
'Data invalida: Data angajarii trebuie sa fie in trecut = ' ||  
to_char( :new.data_angajare, 'YYYY-MM-DD HH24:MI:SS' ) );  
END IF;  
END;  
/
```

```
ALTER TABLE angajat_other  
ADD CONSTRAINT salariu_angajat_check_other  
CHECK (salariu > 0);
```

```

222
223 insert into oras_other
224 values(999, 'Bucuresti', 2);
225
226 insert into oras_other
227 values(999, 'Tafuna', 2);
228
229 --Constrangeri de cheie primară (la nivel local/global)

```

Script Output x Query Result x

Task completed in 0.023 seconds

```

error starting at line : 223 in command -
insert into oras_other
values(999, 'Bucuresti', 2)
error report -
RA-20001: Constangere de unicitate pe nume_oras
         incalcata. Fragmentul de pe bd_ro contine aceeasi valoare
RA-06512: at "MODBD_TRAVEL.TRIG_ORAS_OTHER", line 9
RA-04088: error during execution of trigger 'MODBD_TRAVEL.TRIG_ORAS_OTHER'

```

c) BD_GLOBAL

```

--6. Constrangeri
--iii. unicitate globala pe fragmente verticale

create or replace trigger trig_angajat_unique
instead of insert on angajat
for each row
declare
  nr number(1);
begin

  select count(*) into nr
  from modbd_travel.angajat_ro ar, angajat_other@bd_other ao
  where ar.num = :new.num
  and ao.nr_telefon = :new.nr_telefon;

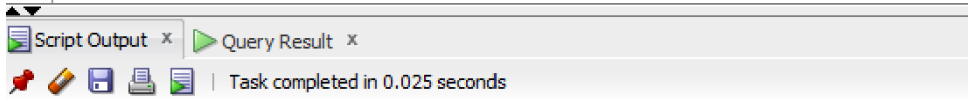
  if (nr<>0) then
    raise_application_error (-20001,'Constangere de unicitate pe nume si nr_telefon
    incalcata. Fragmentele verticale contin aceleasi valori');
  end if;
end;
/

```

```
select * from angajat;
```

```
insert into angajat (id_angajat, nr_telefon, nume)  
values(999, '6782735265', 'Natalie');
```

```
671 |  
672 | insert into angajat (id_angajat, nr_telefon, nume)  
673 | values(999, '6782735265', 'Natalie');  
674 |
```



Error starting at line : 672 in command -

```
insert into angajat (id_angajat, nr_telefon, nume)  
values(999, '6782735265', 'Natalie')
```

Error report -

ORA-20001: Constangere de unicitate pe nume si nr_telefon

incalcata. Fragmentele verticale contin aceleasi valori

ORA-06512: at "MODBD_GLOBAL.TRIG_ANGAJAT_UNIQUE", line 11

ORA-04088: error during execution of trigger 'MODBD_GLOBAL.TRIG_ANGAJAT_UNIQUE'

6. Optimizarea cererii SQL

--BD_RO

--Sa se afiseze rezervarile care au fost efectuate de un angajat cu salariul de cel putin 5000 de lei.

--Pentru aceasta cerinta se pot aplica optimizari bazate pe regula si optimizari bazate pe cost. În general, optimizarea bazată pe cost este considerată mai avansată și mai precisă decât optimizarea bazată pe regulă, deoarece ia în considerare mai mulți factori în procesul de optimizare. Cu toate acestea, optimizarea bazată pe regulă poate fi utilă în anumite situații, cum ar fi atunci când statisticile despre datele din baza de date sunt incomplete sau inexacte.

--a) planul de execuție ales de optimizatorul bazat pe regulă (explicație etape parcurse);

-- setam optimizarea bazata pe regula pentru sesiunea curenta

```
ALTER SESSION
```

```
SET OPTIMIZER_MODE = RULE;
```

-- se rezolva cerinta formulata la Modulul analiza, punctul 9

```
SELECT id_rezervare, data_checkin, data_checkout, pret_total  
FROM rezervare r, angajat_other@bd_other a  
WHERE r.id_angajat = a.id_angajat  
AND a.salariu >= 5000;
```

```

-- se genereaza planul de executie al cererii
EXPLAIN PLAN
SET STATEMENT_ID = 'rbo1'
FOR
SELECT id_rezervare, data_checkin, data_checkout, pret_total
FROM rezervare r, angajat_other@bd_other a
WHERE r.id_angajat = a.id_angajat
AND a.salariu >= 5000;

-- analiza planului de executie obtinut
SELECT plan_table_output FROM
table (dbms_xplan.display('PLAN_TABLE','rbo1','SERIAL'));

```

```

621 SELECT plan_table_output FROM
622 table (dbms_xplan.display('PLAN_TABLE','rbo1','SERIAL'));
623
624

```

PLAN_TABLE_OUTPUT						
5	-----					
6	0	SELECT STATEMENT				
7	1	MERGE JOIN				
8	2	SORT JOIN				
9	3	REMOTE	ANGAJAT_OTHER	BD_OT~	R->S	
10	* 4	SORT JOIN				
11	5	TABLE ACCESS FULL	REZERVARE			
12	-----					
13						
14	Predicate Information (identified by operation id):					
15	-----					
16						
17	4	access("R"."ID_ANGAJAT"="A"."ID_ANGAJAT")				
18		filter("R"."ID_ANGAJAT"="A"."ID_ANGAJAT")				
19						
20	Remote SQL Information (identified by operation id):					
21	-----					
22						
23	3	SELECT "ID_ANGAJAT","SALARIU" FROM "ANGAJAT_OTHER" "A" WHERE				
24		"SALARIU">=5000 (accessing 'BD_OTHER')				
25						
26						
27	Note					
28	-----					
29		- rule based optimizer used (consider using cbo)				

```

--b) planul de execuție ales de optimizatorul bazat pe cost (explicație etape parcurse);
-- se seteaza la nivel de sesiune optimizarea bazata pe cost

```

```

ALTER SESSION
SET OPTIMIZER_MODE = choose;

EXPLAIN PLAN
SET STATEMENT_ID = 'cbo1'
FOR
SELECT id_rezervare, data_checkin, data_checkout, pret_total
FROM rezervare r, angajat_other@bd_other a
WHERE r.id_angajat = a.id_angajat
AND a.salariu >= 5000;

-- analiza planului de executie obtinut
SELECT plan_table_output FROM
table (dbms_xplan.display('PLAN_TABLE','cbo1','SERIAL'));

```

```

638
639 -- analiza planului de executie obtinut
640 SELECT plan_table_output FROM
641 table (dbms_xplan.display('PLAN_TABLE','cbo1','SERIAL'));

```

PLAN_TABLE_OUTPUT									
1	Plan hash value: 1138448718								
2									
3	-----								
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Inst	IN-OUT
5	-----								
6	0	SELECT STATEMENT		139	7228	6 (0)	00:00:01		
7	* 1	HASH JOIN		139	7228	6 (0)	00:00:01		
8	2	REMOTE	ANGAJAT_OTHER	11	275	3 (0)	00:00:01	BD_OT~	R->S
9	3	TABLE ACCESS FULL	REZERVARE	251	6777	3 (0)	00:00:01		
10	-----								
11									
12	Predicate Information (identified by operation id):								
13	-----								
14									
15	1	- access("R"."ID_ANGAJAT"="A"."ID_ANGAJAT")							
16									
17	Remote SQL Information (identified by operation id):								
18	-----								

--c) Pentru a accelera interogarea, se poate crea un index pe coloana salariu.

```

-----
--BD_OTHER

```

ON angajat_other (salariu);

EXPLAIN PLAN

```
SET STATEMENT_ID = 'cbo2'
```

FOR

```
SELECT id_rezervare, data_checkin, data_checkout, pret_total
```

FROM rezervare r, angajat_other@bd_other a

WHERE r.id_angajat = a.id_angajat

AND a.salariu >= 5000;

-- analiza planului de executie obtinut

```
SELECT plan_table_output FROM
```

```
table (dbms_xplan.display('PLAN_TABLE','cbo2','SERIAL'));
```

```
652
653 -- analiza planului de executie obtinut
654 SELECT plan_table_output FROM
655 table (dbsms_xplan.display('PLAN_TABLE','cbo2','SERIAL'));
656
```

```
Script Output x Query Result x
All Rows Fetched: 22 in 0.012 seconds

PLAN_TABLE_OUTPUT
1 Plan hash value: 1138448718
2
3 -----
4 | Id | Operation          | Name              | Rows | Bytes | Cost (%CPU)| Time     | Inst | IN-OUT |
5 -----
6 |  0 | SELECT STATEMENT   |                   |    1 |  300 |    3   (0)| 00:00:01 |      |        |
7 |*  1 |  HASH JOIN         |                   |    1 |  300 |    3   (0)| 00:00:01 |      |        |
8 |  2 |    REMOTE          | ANGAJAT_OTHER     |    12 |   300 |    3   (0)| 00:00:01 | BD_OT~ | R->S |
9 |  3 |      TABLE ACCESS FULL| REZERVARE         |    251 |  6777 |    3   (0)| 00:00:01 |      |        |
10 -----
11
12 Predicate Information (identified by operation id):
13 -----
14
15   1 - access("R"."ID_ANGAJAT"="A"."ID_ANGAJAT")
16
17 Remote SQL Information (identified by operation id):
18 -----
```


III. MODUL FRONT-END

1. Modul aplicație prin care se introduc și gestionează informații la nivelul bazelor de date locale

Pentru fiecare entitate exista un formular prin care se pot introduce date. Pentru serverul bd_ro, vom exemplifica in continuare introducerea unui hotel (relatie fragmentata orizontala) si a unui discount (tabela replicata), vizualizarea hotelurilor din bd_ro, vizualizarea discounturilor, dar si vizualizarea datelor despre angajati (relatie fragmentata verticala) prin intermediul tabelului.

BD-RO TOATE HOTELURILE ADAUGA HOTEL TOATE DISCOUNTURILE ADAUGA DISCOUNT TOTI ANGAJATII

ADAUGA HOTEL

Nume: HOTEL RO1

Numar stele: 2

Capacitate: 100

Selecteaza tip hotel: H

Selecteaza orasul: Bucuresti

ADAUGA HOTEL

Fig 1 – Formular introducerea hotel BD_RO

BD-RO TOATE HOTELURILE ADAUGA HOTEL TOATE DISCOUNTURILE ADAUGA DISCOUNT TOTI ANGAJATII

Lista Hotelurilor

Id Hotel	Nume	Nr stele	Capacitate	Tip hotel	Oras
1	Kellington	3	300	V	Botosani
2	Skeeles	2	420	R	Bucuresti
3	Shelds	1	310	H	Iasi
4	Vaneschi	4	350	M	Sibiu
5	McGaraghan	1	330	V	Cluj
6	Grelik	3	340	P	Brasov
7	Wilshaw	5	390	R	Constanta
8	Aps	4	420	H	Ploiesti
1000	hotel ro1	2	100	H	Bucuresti

1 >>

Fig. 2 – Lista hotelurilor BD_RO

BD-RO
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII

ADAUGA DISCOUNT

Denumire

DISCOUNT 1

Procent

30

Data start

02-MAY-2023

Data final

10-JUN-2023

ADAUGA DISCOUNT

Fig. 3 – Formular introducere dicount in BD_RO

BD-RO
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII

Lista discounturi

Adauga discount

Id discount	Denumire	Procent	Data start	Data final
1	Discount Iunie 2020	30	01-06-2020	01-07-2020
2	Discount Iulie 2022	50	01-07-2022	01-08-2022
3	Discount Ianuarie 2022	40	01-01-2020	01-02-2020
4	Discount Aprilie 2020	50	01-04-2020	01-05-2020
5	Discount Decembrie 2020	40	01-12-2020	01-01-2021
6	Discount Aprilie 2022	10	01-04-2022	01-05-2022
7	Discount Martie 2021	50	01-03-2021	01-04-2021
8	Discount Iunie 2021	10	01-06-2021	01-07-2021
9	Discount Ianuarie 2022	20	01-01-2022	01-02-2022
10	Discount Noiembrie 2021	10	01-11-2021	01-12-2021
1000	discount 1	30	02-05-2023	10-06-2023

1 »

Fig. 4 – Lista discounturi BD_RO

BD-RO	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
Lista angajati					
Id angajat	Nume	Prenume			
1	Natalie	Matteo			
2	Leachman	Rowena			
3	Clancy	Arvy			
4	de Clerk	Herold			
5	Jobling	Lennard			
6	McHenry	Evered			
7	Biddell	Halette			
8	Minihane	Tate			
9	Ivannikov	Allie			
10	Clulee	Elsi			
11	Melan	Netti			
12	Gallemore	Laureen			
13	Sidsaff	Leland			
14	Stiegers	Marilee			
15	Huddles	Diane-marie			

Fig. 5 – Lista angajati BD_RO

BD-OTHER	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
Lista discounturi					
Adauga discount					
Id discount	Denumire	Procent	Data start	Data final	
1	Discount Iunie 2020	30	01-06-2020	01-07-2020	
2	Discount Iulie 2022	50	01-07-2022	01-08-2022	
3	Discount Ianuarie 2022	40	01-01-2020	01-02-2020	
4	Discount Aprilie 2020	50	01-04-2020	01-05-2020	
5	Discount Decembrie 2020	40	01-12-2020	01-01-2021	
6	Discount Aprilie 2022	10	01-04-2022	01-05-2022	
7	Discount Martie 2021	50	01-03-2021	01-04-2021	
8	Discount Iunie 2021	10	01-06-2021	01-07-2021	
9	Discount Ianuarie 2022	20	01-01-2022	01-02-2022	
10	Discount Noiembrie 2021	10	01-11-2021	01-12-2021	
1000	discount 1	30	02-05-2023	10-06-2023	

Fig. 6 – Lista discounturi BD_OTHER

BD-OTHER	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
Lista angajati					
Id angajat	Salariu	Data angajare	Nr telefon		
1	1800	1998-05-02 00:00:00.0	6782735265		
2	3700	1991-10-29 00:00:00.0	9492566846		
3	600	1995-12-31 00:00:00.0	2858446803		
4	9500	1999-04-07 00:00:00.0	3076494005		
5	1200	1991-08-25 00:00:00.0	2981161603		
6	1200	1994-12-21 00:00:00.0	2693830618		
7	2500	1991-01-24 00:00:00.0	4289512482		
8	8700	1993-03-25 00:00:00.0	5071176069		
9	6400	1993-04-08 00:00:00.0	5258077722		
10	5300	1999-08-31 00:00:00.0	2801141423		
11	7400	1997-09-26 00:00:00.0	7738888536		
12	1400	1995-04-30 00:00:00.0	3018996347		
13	6500	1997-11-19 00:00:00.0	2179960647		
14	8900	1994-08-29 00:00:00.0	3209537109		
15	4400	1990-07-08 00:00:00.0	4904309419		

Fig. 7 – Lista angajati BD_OTHER

2. Modul aplicație prin care se vizualizează informații la nivelul bazei de date globale

La nivel global, se pot vizualiza si introduce informatii ce vor fi transmise cu ajutorul triggerilor si bazelor de date locale (bd_ro si bd_other). In continuare voi exemplifica vizualizarea informatiilor la nivelul bazei de date globale cu ajutorul tabelor pentru hoteluri si angajati.

BD-GLOBAL

TOATEHOTELURILE

ADAUGAHOTEL

TOATEDISCOUNTURILE

ADAUGADISCOUNT

TOTIANGAJATII

ADAUGAANGAJAT

Lista Hotelurilor

Id Hotel	Nume	Nr stele	Capacitate	Tip hotel	Oras
16	Alishoner	2	470	H	Escobar
17	Grattan	4	310	V	Kabul
18	Beldom	5	430	R	La Plata
19	Akid	5	490	M	Mogiljov
20	Studman	2	360	P	Molodetno
1000	hotel ro1	2	100	H	Bucuresti

1

2

»

Fig. 8 – Lista hotelurilor BD_GLOBAL

BD-GLOBAL	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII	ADAUGA ANGAJAT
Lista angajati						
Id angajat	Nume	Prenume	Nr. telefon	Data angajare	Salariu	
1	Natalie	Matteo	6782735265	1998-05-02 00:00:00.0	1800	
2	Leachman	Rowena	9492566846	1991-10-29 00:00:00.0	3700	
3	Clancy	Arvy	2858446803	1995-12-31 00:00:00.0	600	
4	de Clerk	Herold	3076494005	1999-04-07 00:00:00.0	9500	
5	Jobling	Lennard	2981161603	1991-08-25 00:00:00.0	1200	
6	McHenry	Evered	2693830618	1994-12-21 00:00:00.0	1200	
7	Biddell	Halette	4289512482	1991-01-24 00:00:00.0	2500	
8	Minihane	Tate	5071176069	1993-03-25 00:00:00.0	8700	
9	Ivannikov	Allie	5258077722	1993-04-08 00:00:00.0	6400	
10	Clulee	Elsi	2801141423	1999-08-31 00:00:00.0	5300	
11	Melan	Netti	7738888536	1997-09-26 00:00:00.0	7400	
12	Gallimore	Laureen	3018996347	1995-04-30 00:00:00.0	1400	
13	Sidsaff	Leland	2179960647	1997-11-19 00:00:00.0	6500	
14	Stiegers	Marilee	3209537109	1994-08-29 00:00:00.0	8900	
15	Huddles	Diane-marie	4904309419	1990-07-08 00:00:00.0	4400	

Fig. 9 – Lista angajati BD_GLOBAL

- Posibilitatea de vizualizare la nivelul bazei de date globale a modificărilor aplicate asupra datelor stocate în bazele de date locale (efectele operațiilor LMD locale realizate pe fragmentele orizontale, pe fragmentele verticale, respectiv pe relațiile replicate).

Pentru a vizualiza propagarea modificarilor aplicate asupra datelor stocate in bazele de date locale pentru o relatie fragmentata orizontala, vom folosi relatia hotel si vom introduce in baza de date locala bd_other hotelul cu numele "HOTEL OTHER 1" si id-ul 1001 si in baza de date locala bd_ro hotelul cu numele "HOTEL RO 2" si id-ul 1002. In Fig. 12 se poate observa ca hotelurile introduse in bazele de date locale sunt vizibile si la nivelul bazei de date globale (bd_global).

BD-GLOBAL

TOATEHOTELURILE

ADAUGAHOTEL

TOATEDISCOUNTURILE

ADAUGADISCOUNT

TOTIANGAJATII

ADAUGAANGAJAT

Lista Hotelurilor

Id Hotel	Nume	Nr stele	Capacitate	Tip hotel	Oras
16	Alishoner	2	470	H	Escobar
17	Grattan	4	310	V	Kabul
18	Beldom	5	430	R	La Plata
19	Akid	5	490	M	Mogiljov
20	Studman	2	360	P	Molodetno
1000	hotel ro1	2	100	H	Bucuresti

1

2

»

Fig. 10 – Lista hotelurilor BD_GLOBAL
(inainte de introducerea datelor in bazele de date locale)

BD-OTHER	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
-----------------	-----------------------------	-------------------------	--------------------------------	----------------------------	---------------------------

ADAUGA HOTEL

Nume

HOTEL OTHER 1

Numar stele

2

Capacitate

1000

Selecteaza tip hotel

P

Selecteaza orasul

Tafuna

ADAUGA HOTEL

Fig. 11 – Adaugarea unui hotel in BD_OTHER

BD-GLOBAL

TOATE HOTELURILE

ADAUGA HOTEL

TOATE DISCOUNTURILE

ADAUGA DISCOUNT

TOTI ANGAJATII

ADAUGA ANGAJAT

Lista Hotelurilor

Id Hotel	Nume	Nr stele	Capacitate	Tip hotel	Oras
16	Alishoner	2	470	H	Escobar
17	Grattan	4	310	V	Kabul
18	Beldom	5	430	R	La Plata
19	Akid	5	490	M	Mogiljov
20	Studman	2	360	P	Molodetno
1000	hotel ro1	2	100	H	Bucuresti
1001	hotel other 1	2	1000	P	Tafuna
1002	hotel ro2	3	200	R	Iasi

1

2

>>

Fig. 12 – Lista hotelurilor BD_GLOBAL
(dupa introducerea datelor in bazele de date locale)

Pentru a vizualiza propagarea modificarilor aplicate asupra datelor stocate in bazele de date locale pentru o tabela replicata, vom folosi relatia discount si vom introduce in baza de date locala bd_ro discountul cu numele “DISCOUNT 1” si id-ul 1000. In Fig. 14 se poate observa ca discountul introdus in baza de date locala bd_ro este vizibil si la nivelul bazei de date globale (bd_global).

BD-RO	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
-------	------------------	--------------	---------------------	-----------------	----------------

ADAUGA DISCOUNT

Denumire	Procent
DISCOUNT 1	30
Data start	Data final
02 - MAY - 2023	10 - JUN - 2023

ADAUGA DISCOUNT

Fig. 13 – Adaugarea unui discount in BD_RO

BD-GLOBAL

TOATEHOTELURILE

ADAUGAHOTEL

TOATEDISCOUNTURILE

ADAUGADISCOUNT

TOTIANGAJATII

ADAUGAANGAJAT

Lista discounturi

Adauga discount

Id discount	Denumire	Procent	Data start	Data final
1	Discount Iunie 2020	30	01-06-2020	01-07-2020
2	Discount Iulie 2022	50	01-07-2022	01-08-2022
3	Discount Ianuarie 2022	40	01-01-2020	01-02-2020
4	Discount Aprilie 2020	50	01-04-2020	01-05-2020
5	Discount Decembrie 2020	40	01-12-2020	01-01-2021
6	Discount Aprilie 2022	10	01-04-2022	01-05-2022
7	Discount Martie 2021	50	01-03-2021	01-04-2021
8	Discount Iunie 2021	10	01-06-2021	01-07-2021
9	Discount Ianuarie 2022	20	01-01-2022	01-02-2022
10	Discount Noiembrie 2021	10	01-11-2021	01-12-2021
1000	discount 1	30	02-05-2023	10-06-2023

1

»

Fig. 14 – Lista discounturi BD_GLOBAL
(dupa introducerea discount 1000 in bd_ro)

- Posibilitatea de verificare la nivelul bazelor de date locale a propagării operațiilor LMD realizate la nivelul bazei de date globale (efectele operațiilor LMD realizate la nivel global ce se vor reflecta asupra fragmentelor orizontale, a fragmentelor verticale, respectiv a relațiilor replicate).

Pentru a vizualiza propagarea modificarilor aplicate asupra datelor stocate in baza de date globala pentru o relatie fragmentata vertical, vom folosi relatia angajat si vom introduce in baza de date globala bd_global angajatul cu id-ul „1000”. Aceste date vor fi vizibile in bazele de date locale conform fragmentarii verticale: datele sensibile precum nr_telefon, salariu si data_angajare sunt prezente in bd_other, iar celelalte date, nume si prenume, sunt prezente in bd_ro. In Fig. 17 si Fig. 18 se poate observa ca datele despre angajatul introdus in baza de date globala bd_global sunt prezente si in bazele de date locale conform fragmentarii verticale.

BD-GLOBAL
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII
ADAUGA ANGAJAT

ADAUGA ANGAJAT

Nume
ANGAJAT1

Prenume
PRENUME1

Numar telefon
07339877654

Data angajare
03-MAY-2023

Salariu
3000

ADAUGA ANGAJAT

Fig. 15 – Adaugare angajat BD_GLOBAL

BD-GLOBAL
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII
ADAUGA ANGAJAT

Lista angajati

Id angajat	Nume	Prenume	Nr. telefon	Data angajare	Salariu
16	Schober	Natalya	8256874453	1990-11-21 00:00:00.0	3000
17	Estevez	Melisenda	2347137505	1993-05-26 00:00:00.0	5400
18	Lemme	Jennifer	9407441219	1999-03-20 00:00:00.0	6600
19	Baldree	Lucie	2108438048	1992-03-05 00:00:00.0	6800
20	Prayer	Shane	7289149194	1995-01-13 00:00:00.0	2000
1000	angajat1	prenume1	07339877654	2023-05-03 00:00:00.0	3000

1
2
»

Fig. 16 – Lista angajati BD_GLOBAL
(dupa introducere angajat 1000)

BD-RO	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
Lista angajati					
Id angajat	Nume	Prenume			
16	Schober	Natalya			
17	Estevez	Melisenda			
18	Lemme	Jennifer			
19	Baldree	Lucie			
20	Prayer	Shane			
1000	angajat1	prenume1			
			1	2	»

Fig. 17 – Lista angajati BD_RO
(dupa introducere angajat 1000 in BD_GLOBAL)

BD-OTHER	TOATE HOTELURILE	ADAUGA HOTEL	TOATE DISCOUNTURILE	ADAUGA DISCOUNT	TOTI ANGAJATII
Lista angajati					
Id angajat	Salariu	Data angajare	Nr telefon		
16	3000	1990-11-21 00:00:00.0	8256874453		
17	5400	1993-05-26 00:00:00.0	2347137505		
18	6600	1999-03-20 00:00:00.0	9407441219		
19	6800	1992-03-05 00:00:00.0	2108438048		
20	2000	1995-01-13 00:00:00.0	7289149194		
1000	3000	2023-05-03 00:00:00.0	07339877654		
			1	2	»

Fig. 18 – Lista angajati BD_OTHER
(dupa introducere angajat 1000 in BD_GLOBAL)

Pentru a vizualiza propagarea modificarilor aplicate asupra datelor stocate in baza de date globala pentru o relatie fragmentata orizontala, vom folosi relatia hotel si vom introduce in baza de date globala bd_global hotelul cu id-ul „1002” ce este localizat intr-un oras din Romania. Datele hotelului vor fi vizibile in baza de date locala bd_ro conform fragmentarii orizontale: hotelurile ce sunt localizate in orase din Romania vor fi prezente in bd_ro. In Fig. 20 se poate observa ca datele despre hotelul introdus in baza de date globala bd_global s-au propagat si in baza de date locala bd_ro.

BD-GLOBAL
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII
ADAUGA ANGAJAT

ADAUGA HOTEL

Nume

HOTEL RO2

Numar stele

3

Capacitate

200

Selecteaza tip hotel

R

Selecteaza orasul

Iasi

ADAUGA HOTEL

Fig. 19 – Adaugare hotel BD_GLOBAL

BD-RO
TOATE HOTELURILE
ADAUGA HOTEL
TOATE DISCOUNTURILE
ADAUGA DISCOUNT
TOTI ANGAJATII

Lista Hotelurilor

Id Hotel	Nume	Nr stele	Capacitate	Tip hotel	Oras
1	Kellington	3	300	V	Botosani
2	Skeeles	2	420	R	Bucuresti
3	Sheilds	1	310	H	Iasi
4	Vaneschi	4	350	M	Sibiu
5	McGaraghan	1	330	V	Cluj
6	Greluk	3	340	P	Brasov
7	Wilshaw	5	390	R	Constanta
8	Aps	4	420	H	Ploiesti
1000	hotel ro1	2	100	H	Bucuresti
1002	hotel ro2	3	200	R	Iasi

1
»

Fig. 20 – Lista hoteluri BD_RO
(dupa introducere hotel 1002 in BD_GLOBAL)