## TAMBook - Librărie online

## Descrierea modelului ales și a obiectivelor aplicației

TAMBook este o aplicație internă care îi permite administratorului să înregistreze o comandă pe care vrea să o plaseze utilizatorul care i-a cerut aceasta ce va conține cărțile alese în cantitatea dorită. De asemenea, dintre funcționalitățile aplicației se enumeră: vizualizarea utilizatorilor, detaliilor pentru o comandă dată, vizualizarea cărților, comenzilor și a detaliilor acestora, recenziilor, adreselor, orașelor, țărilor. Mai mult, aplicația permite modificarea numelor țărilor, orașelor (în cazul în care este necesar); utilizatorul își poate modifica numele, poate modifica cantitatea cărților dintr-o comandă, poate modifica numele străzii și blocului pentru o adresă deja înregistrată, pot modifica mesajul din review și ratingul oferit. De asemenea, mai poate fi modificat prețul cărții.

Astfel, un utilizator poate avea mai multe adrese care conțin date de livrare printre care și orașul care face parte dintr-o țară (o țară are mai multe orașe, iar un oraș poate fi inclus în mai multe adrese), iar acesta poate plasa mai multe comenzi la adrese diferite.

În ceea ce privește comanda, aceasta, inițial, se creează pentru un utilizator și adresa selectată, cu totalul pe 0, iar în timp ce sunt adăugate detaliile comenzii – cartea aleasă, cu prețul ei în momentul respectiv, cantitatea și discountul dacă este cazul – atunci totalul din comandă este actualizat pentru a corespunde cu realitatea.

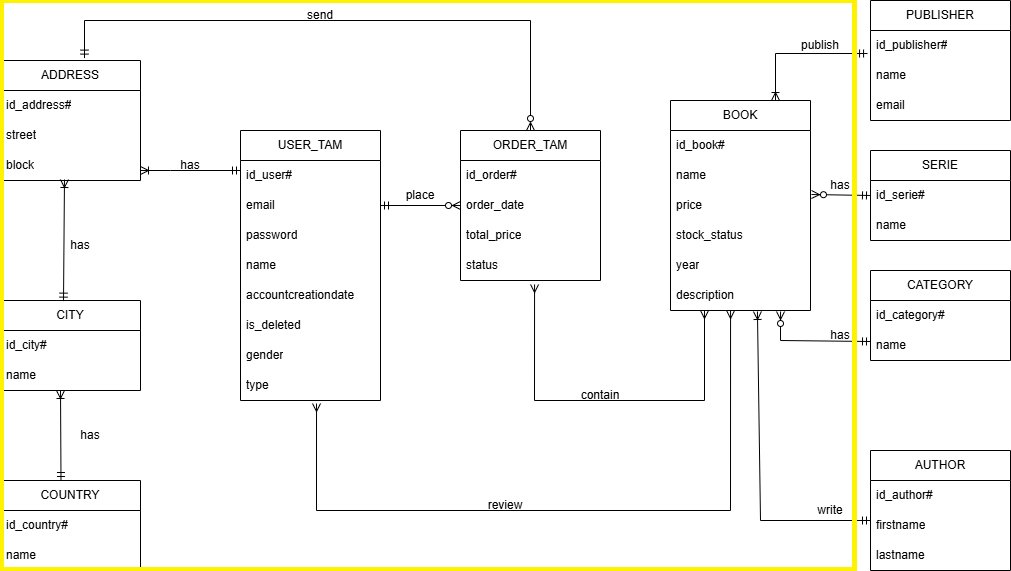
Mai mult, în acest model, o carte este publicată de o singură editură, are o singură serie, o singură categorie și un singur autor,dar o serie, categorie, autor, editură pot avea mai multe cărți. Totodată, un utilizator poate scrie mai multe recenzii pentru cărțile dorite, chiar și mai multe în aceeași zi.

Aplicația are ca scop managementul unei librării online, prin gestionarea produselor și vânzărilor, dorindu-se observarea cărților care sunt cel mai bine vândute. Aplicația permite ștergerea utilizatorilor, comenzilor, detaliilor despre comandă, adreselor, țărilor, orașelor, recenziilor.

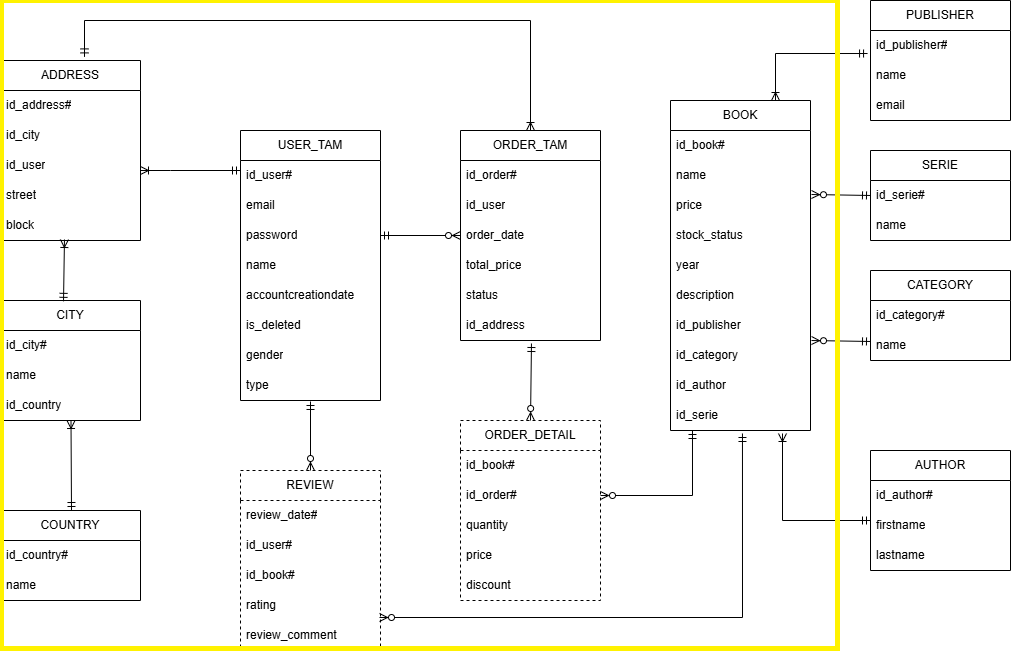
Observație: S-a stabilit la consultații să nu se mai implementeze tabelele publisher, serie, category, author.

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

1. Diagrama entitate – relație a bazei de date OLTP inițiale



b. Diagrama conceptuală a bazei de date OLTP inițiale



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

Aplicația are la bază două servere de baze de date numite:

* orcl
* orcl2

Utilizatorii bazei de date orcl sunt:

1. Utilizatorul pentru baza de date centralizată - **DWBI\_miruna** (pentru baza de date oltp; name: DWBI\_miruna)
2. Utilizatorul care poate gestiona date din afara Europei - **bdd** (name: db1\_bdd\_exeu)
3. Utilizatorul global - **bdd\_global** (name: db1\_bdd\_global\_tam); acesta gestionează atât date din Europa, cât și din afara ei și stochează fragmentul cu date sensibile ale relației **USER\_TAM**

Utilizatorii bazei de date orcl2 sunt:

1. Utilizatorul care poate gestiona date din Europa - **bdd** (name: db2\_bdd\_eu\_tam)

Pentru comunicarea dintre serverul de baze de date orcl și serverul de baze de date orcl2 s-au creat următoarele două link-uri:

* orcl -> orcl2: bd\_eu
* orcl2 -> orcl: bd\_exeu

## 4. **Argumentarea deciziei de fragmentare a relațiilor**

1. **Fragmentare orizontală primară**

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

(TARILE actuale din bd: Anglia, Australia, Canada, Estonia, Franta, Germania, India, Italia, Macedonia, Romania, SUA, Spania, Turcia, Ucraina)

Fragmentare orizontală PRIMARĂ - relația COUNTRY

**După analiza celor mai utilizate cereri cele mai folosite predicate sunt p1, p2.**

p1: reprezintă țările din Europa (country.name IN Estonia, Franta, Germania, Italia, Macedonia, Romania, Spania, Ucraina)

p2: reprezintă țările din afara Europei (country.name IN Anglia, Australia, Canada, India, SUA, Turcia)

Aceste două predicate sunt cele mai utilizate în aplicație (adică este des utilizat doar partea de Europa și doar partea din afara Europei)

Pr = {p1,p2} mulțimea de predicate simple

Pr este COMPLETĂ, deoarece probabilitățile de acces ale fiecărei aplicații la orice tuplu care aparține oricărui fragment orizontal definit pe baza acestei mulțimi sunt egale.

Pr este MINIMALĂ, deoarece toate predicatele mulțimii sunt relevante.

Mulțimea predicatelor compuse - M

ȚĂRILE actuale din baza de date centralizată (oltp): (din Europa) Estonia, Franta, Germania, Italia, Macedonia, Romania, Spania, Ucraina; (din afara Europei) Anglia, Australia, Canada, India, SUA, Turcia

Fie Europa = country.name IN Estonia, Franta, Germania, Italia, Macedonia, Romania, Spania, Ucraina

Fie ExEuropa = country.name IN Anglia, Australia, Canada, India, SUA, Turcia

Determinarea mulțimii M pe baza Pr = {Europa, ExEuropa}

Determinarea mulțimii I de implicații:

i1: (country.name = Europa) => ￢(country.name = ExEuropa)

i2: ￢(country.name = Europa) => (country.name = ExEuropa)

Astfel, mulțimea M va avea 4 predicate compuse:

m1: (country.name = Europa) ∧ (country.name = ExEuropa)

m2: (country.name = Europa) ∧ ￢(country.name = ExEuropa)

m3: ￢(country.name = Europa) ∧ (country.name = ExEuropa)

m4: ￢(country.name = Europa) ∧ ￢(country.name = ExEuropa)

Din cele 4 predicate se observă că m1 și m4 nu au sens, având în vedere implicațiile de mai sus, ceea ce duce la eliminarea acestora.

Se observă că m2 este echivalent cu p1, iar m3 cu p2.

Astfel, M = {m2,m3} = { p1,p2} = Pr

Fragmentarea este corectă dacă îndeplinește regulile de completitudine, reconstrucție și disjuncție.

Completitudine: M este completă

Reconstrucția: Country\_eu ∪ Country\_exeu = Country

Disjuncția: Country\_eu ∩ Country\_exeu = ∅

Astfel, fragmentarea este corectă.

**ii. Obținerea fragmentelor orizontale primare**

* stația 1 (orcl): COUNTRY\_EXEU
* stația 2 (orcl2): COUNTRY\_EU

b. **Fragmentare orizontală derivată**

i.Obținerea fragmentelor orizontale derivate

Fragmentarea orizontală primară a relației COUNTRY va duce la fragmentarea orizontală derivată a relațiilor: CITY, ADDRESS, ORDER\_TAM și ORDER\_DETAIL.

Aceste relații vor fi fragmentate pornind de la predicatele

p1: Europa

p2: ExEuropa

Stocarea fragmentelor se va face astfel:

* stația 1 (orcl): CITY\_EXEU, ADDRESS\_EXEU, ORDER\_TAM\_EXEU, ORDER\_DETAIL\_EXEU
* stația 2 (orcl2): CITY\_EU, ADDRESS\_EU, ORDER\_TAM\_EU, ORDER\_DETAIL\_EU

c. **Fragmentarea verticală**

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

Relația căreia îi va fi aplicată fragmentarea verticală este USER\_TAM(id\_user#, email, password, name, accountcreationdate, is\_deleted, gender, type)

Cele mai des utilizate cereri sunt următoarele

q1: Afișați genul clienților de tip juridic care și-au creat cont în ultimele 6 luni și sunt activi.

SELECT gender

FROM USER\_TAM

WHERE accountcreationdate >= add\_months( sysdate, -6 )

AND type = ‘Persoana juridica’

AND is\_deleted = false;

q2: Autentificare utilizator  
SELECT name

FROM USER\_TAM

WHERE email = ‘ana@yahoo.com’

AND password= ‘pppp’;

q3: Să se afle data creării contului utilizatorilor de gen feminin care au contul inactiv

SELECT accountcreationdate

FROM USER\_TAM

WHERE gender = ‘F’ AND is\_deleted = true;

q4: Să se afișeze numele și email-ul pentru utilizatorii care au contul activ

SELECT email, name

FROM USER\_TAM

WHERE is\_deleted = false;

—----------------------------------------------------------------------------------------------------

Date sensibile:gender, accountcreationdate, is\_deleted, type

Date normale: email, name, password

=> USER\_TAM\_GLOBAL (id\_user#,is\_deleted, accountcreationdate, gender, type)

=> USER\_TAM\_EXEU (id\_user#, name, email, password)

Attribute

A1 = email

A2 = password

A3 = name

A4 = accountcreationdate

A5 = is\_deleted

A6 = gender

A7 = type

Presupunem că numărul de accesări ale atributelor (Ai, Aj) pentru cele 4 cereri de pe cele două stații este 1. Astfel, nr\_accl(qk) = 1, ∀k ∊ {1,2,3,4}, ∀l ∊ {1,2}

**Frecvența de acces** a cererii qk de pe stația Sl se notează fr\_accl(qk)

|  |  |  |
| --- | --- | --- |
| Stații | Stația 1 | Stația 2 |
| q1 | 20 | 5 |
| q2 | 5 | 10 |
| q3 | 15 | 5 |
| q4 | 1 | 10 |

**Matricea VA**

VA(i,j) = ref(qi, Aj) = 1, dacă Aj este referit în qi

= 0, altfel

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| VA | A1 | A2 | A3 | A4 | A5 | A6 | A7 |
| q1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| q2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| q3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| q4 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

**Matricea afinității atributelor (AA)**

af(Ai, Aj) = nr\_accl(qk) fr\_accl(qk), K = { k | ref(qk, Ai)=1 ⋀ ref(qk, Aj)=1}

af(A1,A1) = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 5 + 10 + 1 + 10 = 26

af(A1,A2) = fr\_acc1(q2) + fr\_acc2(q2) = 5 + 10 = 15

af(A1,A3) = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

af(A1,A4) = 0

af(A1,A5) = fr\_acc1(q4) + fr\_acc2(q4) = 1 + 10 = 11

af(A1,A6) = af(A1,A7) = 0

af(A2,A2) = fr\_acc1(q2) + fr\_acc2(q2) = 15

af(A2,A3) = fr\_acc1(q2) + fr\_acc2(q2) = 15

af(A2,A4) = af(A2,A5) = af(A2,A6) = af(A2, A7) = 0

af(A3,A3) = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

af(A3,A4) = 0

af(A3,A5) = fr\_acc1(q4) + fr\_acc2(q4) = 11

af(A3,A6) = af(A3,A7) = 0

af(A4,A4) = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 20 + 5 + 15 + 5 = 45

af(A4,A5) = af(A4,A6) = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

af(A4,A7) = fr\_acc1(q1) + fr\_acc2(q1) = 20 + 5 = 25

af(A5,A5) = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) + fr\_acc1(q4) + fr\_acc2(q4) = 45 + 11 = 56

af(A5,A6) = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

af(A5,A7) = fr\_acc1(q1) + fr\_acc2(q1) = 25

af(A6,A6) = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

af(A6,A7) = fr\_acc1(q1) + fr\_acc2(q1) = 25

af(A7,A7) = fr\_acc1(q1) + fr\_acc2(q1) = 25

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AA | A1 | A2 | A3 | A4 | A5 | A6 | A7 |
| A1 | 26 | 15 | 26 | 0 | 11 | 0 | 0 |
| A2 | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| A3 | 26 | 15 | 26 | 0 | 11 | 0 | 0 |
| A4 | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| A5 | 11 | 0 | 11 | 45 | 56 | 45 | 25 |
| A6 | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| A7 | 0 | 0 | 0 | 25 | 25 | 25 | 25 |

1. Gruparea atributelor cu afinități mari

Matricea legăturilor de afinitate (LA)

cont(Ai,Ak,Aj) = 2leg(Ai,Ak) + 2leg(Ak,Aj) - 2leg(Ai,Aj)

leg(Ax, Ay) = af(Ai,Ax) af(Ai,Ay)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 15 | 0 | 0 | 0 | 0 | 0 |
| **2** | 15 | 15 | 0 | 0 | 0 | 0 | 0 |
| **3** | 26 | 15 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 11 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

index = 3

i=1,2 acum i=1

cont(A0,A3,A1) = 2\* af(Ai,A0)af(Ai,A3) + 2\* af(Ai,A3)af(Ai,A1) - 2\* af(Ai,A0)af(Ai,A1) = 2\*(26\*26+15\*15+26\*26+11\*11) = 2\*(676+225+676+121) = 2\*1698 = **3396**

1=2

cont(A1,A3,A2) = 2\* af(Ai,A1)af(Ai,A3) + 2\* af(Ai,A3)af(Ai,A2) - 2\* af(Ai,A1)af(Ai,A2) = 3396 + 2\*(26\*15+ 15\*15+26\*15) - 2\*(26\*15+ 15\*15+26\*15) = **3396**

cont(A2,A3,A4) = 2\* af(Ai,A2)af(Ai,A3) + 2\* af(Ai,A3)af(Ai,A4) - 2\* af(Ai,A2)af(Ai,A4) = 2 \* (15\*26+15\*15+15\*26) + 2\*11\*45 - 2\*0 =2\*(390+225+390) + 22\*45 = 2010 + 990 = 3000

loc = 2

loc = 1

A3 A1 A2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 26 | 15 | 0 | 0 | 0 | 0 |
| **2** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **3** | 26 | 26 | 15 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 11 | 11 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

index = 4

i = 1-3, i=1

cont(A0,A4,A1) = 2\* af(Ai,A0)af(Ai,A4) + 2\* af(Ai,A4)af(Ai,A1) - 2\* af(Ai,A0)af(Ai,A1)

= 2 \* 11 \* 45 = 990

i=2

cont(A1,A4,A2) = 2\* af(Ai,A1)af(Ai,A4) + 2\* af(Ai,A4)af(Ai,A2) - 2\* af(Ai,A1)af(Ai,A2) = 990 - 2010 = -1020

i=3

cont(A2,A4,A3) = 2\* af(Ai,A2)af(Ai,A4) + 2\* af(Ai,A4)af(Ai,A3) - 2\* af(Ai,A2)af(Ai,A3)

= 0 + 990 - 2010 = -1020

cont(A3,A4,A5) = 2\* af(Ai,A3)af(Ai,A4) + 2\* af(Ai,A4)af(Ai,A5) - 2\* af(Ai,A3)af(Ai,A5)

= 990 + 2\*(45\*45+45\*56+45\*45+25\*25) - 2376 = 990 + 14390 - 2376 = **13004**

loc = 4

A3 A1 A2 A4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 26 | 15 | 0 | 0 | 0 | 0 |
| **2** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **3** | 26 | 26 | 15 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 45 | 0 | 0 | 0 |
| **5** | 11 | 11 | 0 | 45 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 45 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 25 | 0 | 0 | 0 |

index = 5

i = 1

cont(A0,A5,A1) = 2\* af(Ai,A0)af(Ai,A5) + 2\* af(Ai,A5)af(Ai,A1) - 2\* af(Ai,A0)af(Ai,A1) = 2\*(11\*26+11\*26+56\*11) = 2376

i=2

cont(A1,A5,A2) = 2\* af(Ai,A1)af(Ai,A5) + 2\* af(Ai,A5)af(Ai,A2) - 2\* af(Ai,A1)af(Ai,A2) = 2376 + 2\*(11\*15+11\*15) - 2010 = 2376 + 660 -2010 = 1026

i = 3

cont(A2,A5,A3) = 2\* af(Ai,A2)af(Ai,A5) + 2\* af(Ai,A5)af(Ai,A3) - 2\* af(Ai,A2)af(Ai,A3) = 660 + 2376 - 2010 = 1026

i=4

cont(A3,A5,A4) = 2\* af(Ai,A3)af(Ai,A5) + 2\* af(Ai,A5)af(Ai,A4) - 2\* af(Ai,A3)af(Ai,A4)

= 2376 + 14390 - 990 = **15776**

cont(A4,A5,A6)= 2\* af(Ai,A4)af(Ai,A5) + 2\* af(Ai,A5)af(Ai,A6) - 2\* af(Ai,A4)af(Ai,A6)

= 14390 + 14390 - 13400 = 15380

loc = 4,

A3,A1,A2,A5,A4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **2** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **3** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 45 | 45 | 0 | 0 |
| **5** | 11 | 11 | 0 | 56 | 45 | 0 | 0 |
| **6** | 0 | 0 | 0 | 45 | 45 | 0 | 0 |
| **7** | 0 | 0 | 0 | 25 | 25 | 0 | 0 |

index = 6

i=1-5

i=1

cont(A0,A6,A1)=2\* af(Ai,A0)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A1) - 2\* af(Ai,A0)af(Ai,A1) = 990

i = 2

cont(A1,A6,A2)=2\* af(Ai,A1)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A2) - 2\* af(Ai,A1)af(Ai,A2) = 990 + 0 - 2010 = -1020

i = 3

cont(A2,A6,A3)=2\* af(Ai,A2)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A3) - 2\* af(Ai,A2)af(Ai,A3) = 0+ 990 - 2010 = -1020

i = 4

cont(A3,A6,A4)=2\* af(Ai,A3)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A4) - 2\* af(Ai,A3)af(Ai,A4) = 990 + 13400 - 990 = 13400

i = 5

cont(A4,A6,A5)=2\* af(Ai,A4)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A5) - 2\* af(Ai,A4)af(Ai,A5)

= 13400 + 14390 - 14390 = 13400

cont(A5,A6,A7)=2\* af(Ai,A5)af(Ai,A6) + 2\* af(Ai,A6)af(Ai,A7) - 2\* af(Ai,A5)af(Ai,A7) = 14390 + 8000 - 4275 = **18115**

loc = 6

A3,A1,A2,A5,A4,A6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **2** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **3** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 45 | 45 | 45 | 0 |
| **5** | 11 | 11 | 0 | 56 | 45 | 45 | 0 |
| **6** | 0 | 0 | 0 | 45 | 45 | 45 | 0 |
| **7** | 0 | 0 | 0 | 25 | 25 | 25 | 0 |

index = 7

i = 1,6

i = 1

cont(A0,A7,A1)=2\* af(Ai,A0)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A1) - 2\* af(Ai,A0)af(Ai,A1) = 550

i = 2

cont(A1,A7,A2)=2\* af(Ai,A1)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A2) - 2\* af(Ai,A1)af(Ai,A2) = 550 + 0 - 2010 = -1460

i = 3

cont(A2,A7,A3)=2\* af(Ai,A2)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A3) - 2\* af(Ai,A2)af(Ai,A3) = 0 + 550 - 2010 = -1460

i = 4

cont(A3,A7,A4)=2\* af(Ai,A3)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A4) - 2\* af(Ai,A3)af(Ai,A4) =550 + 8000 - 990 = 7560

i = 5

cont(A4,A7,A5)=2\* af(Ai,A4)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A5) - 2\* af(Ai,A4)af(Ai,A5) =8000 + 4275 - 14390 = -2115

i = 6

cont(A5,A7,A6)=2\* af(Ai,A5)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A6) - 2\* af(Ai,A5)af(Ai,A6) = 4275 + 8000 - 14390 = -2115

cont(A6,A7,A8) = 2\* af(Ai,A6)af(Ai,A7) + 2\* af(Ai,A7)af(Ai,A8) - 2\* af(Ai,A6)af(Ai,A8) = **8000**

loc = 7

A3,A1,A2,A5,A4,A6,A7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **1** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **2** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **3** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| **5** | 11 | 11 | 0 | 56 | 45 | 45 | 25 |
| **6** | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| **7** | 0 | 0 | 0 | 25 | 25 | 25 | 25 |

Ordonarea liniilor conform ordinii stabilite pentru coloane => Liniile au fost puse în ordinea 3,1,2,5,4,6,7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CA** | **1 (A3)** | **2 (A1)** | **3 (A2)** | **4 (A5)** | **5 (A4)** | **6 (A6)** | **7 (A7)** |
| **1** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **2** | 26 | 26 | 15 | 11 | 0 | 0 | 0 |
| **3** | 15 | 15 | 15 | 0 | 0 | 0 | 0 |
| **4** | 11 | 11 | 0 | 56 | 45 | 45 | 25 |
| **5** | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| **6** | 0 | 0 | 0 | 45 | 45 | 45 | 25 |
| **7** | 0 | 0 | 0 | 25 | 25 | 25 | 25 |

Partiția 1:

TA = {A3}

BA = {A1,A2,A5,A4,A6,A7}

Aplicații care accesează doar A3

TQ\_1 = ⦰

Aplicații care accesează doar A1,A2,A5,A4,A6,A7 FĂRĂ A3

BQ\_1 = {q1, q3}

OQ\_1= {q2,q4}

CTQ\_1 = 0

CBQ\_1 = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

COQ\_1 = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

z\_1 = CTQ\_1 \* CBQ\_1 - COQ\_1\*COQ\_1 = - 26\*26 = - 676

Partiția 2:

TA = {A3,A1}

BA = {A2,A5,A4,A6,A7}

Aplicații care accesează doar A3, A1

TQ\_2 = ⦰

Aplicații care accesează doar A2,A5,A4,A6,A7 FĂRĂ A3, A1

BQ\_2 = {q1, q3}

OQ\_2= {q2,q4}

CTQ\_2 = 0

CBQ\_2 = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

COQ\_2 = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

z\_2 = CTQ\_2 \* CBQ\_2 - COQ\_2\*COQ\_2 = - 26\*26 = - 676

Partiția 3:

TA = {A3,A1, A2}

BA = {A5,A4,A6,A7}

Aplicații care accesează doar A3, A1, A2

TQ\_3 = {q2}

Aplicații care accesează doar A5 A4 A6 A7 FĂRĂ A3, A1, A2

BQ\_3 = {q1, q3}

OQ\_3= {q4}

CTQ\_3 = fr\_acc1(q2) + fr\_acc2(q2) = 15

CBQ\_3 = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

COQ\_3 = fr\_acc1(q4) + fr\_acc2(q4) = 11

z\_3 = CTQ\_3 \* CBQ\_3 - COQ\_3\*COQ\_3 = 15\*45 - 11\*11 = 675 - 121 = 554

Partiția 4:

TA = {A3,A1,A2,A5}

BA = {A4,A6,A7}

Aplicații care accesează doar A3, A1, A2,A5

TQ\_4 = {q2, q4}

Aplicații care accesează doar A4,A6,A7 FĂRĂ A3, A1, A2, A5

BQ\_4 = ⦰

OQ\_4= {q1,q3}

CTQ\_4 = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

CBQ\_4 = 0

COQ\_4 = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

z\_4 = CTQ\_4 \* CBQ\_4 - COQ\_4\*COQ\_4 = - 45 \* 45 = - 2025

Partiția 5:

TA = {A3,A1, A2, A5,A4}

BA = {A6,A7}

Aplicații care accesează doar A3, A1, A2,A5,A4

TQ\_5 = {q2, q4}

Aplicații care accesează doar A6,A7 FĂRĂ A3, A1, A2, A5, A4

BQ\_5 = ⦰

OQ\_5= {q1,q3}

CTQ\_5 = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) = 26

CBQ\_5 = 0

COQ\_5 = fr\_acc1(q1) + fr\_acc2(q1) + fr\_acc1(q3) + fr\_acc2(q3) = 45

z\_5 = CTQ\_5 \* CBQ\_5 - COQ\_5\*COQ\_5 = - 45 \* 45 = - 2025

Partiția 6:

TA = {A3,A1, A2, A5,A4, A6}

BA = {A7}

Aplicații care accesează doar A3, A1, A2,A5,A4,A6

TQ\_6 = {q2, q3, q4}

Aplicații care accesează doar A7 FĂRĂ A3, A1, A2, A5, A4, A6

BQ\_6 = ⦰

OQ\_6= {q1}

CTQ\_6 = fr\_acc1(q2) + fr\_acc2(q2) + fr\_acc1(q4) + fr\_acc2(q4) + fr\_acc1(q3) + fr\_acc2(q3) = 46

CBQ\_6 = 0

COQ\_6 = fr\_acc1(q1) + fr\_acc2(q1) = 25

z\_6 = CTQ\_6 \* CBQ\_6 - COQ\_6\*COQ\_6 = - 25\*25 = -625

Astfel, valoarea cea mai mare pentru z este z\_3 = 554 (partiția 3)

Partiția 3:

TA = {A3,A1, A2} = {name, email, password}

BA = {A5,A4,A6,A7} = {is\_deleted, accountcreationdate, gender, type}

La aceste fragmente se adaugă cheia primară.

Astfel, cele două fragmente sunt:

{id\_user#, name,email, password} și {id\_user#, is\_deleted, accountcreationdate, gender, type}

ii. Obținerea fragmentelor verticale

Astfel, fragmentele obținute

=> USER\_TAM\_GLOBAL (id\_user#, is\_deleted, accountcreationdate, gender, type)

=> USER\_TAM\_EXEU (id\_user#, name, email, password)

## 5 .**Verificarea corectitudinii fragmentărilor realizate**

Având în vedere verificările realizate în partea de implementare pentru fragmentările orizontale primară, derivate și fragmentarea verticală se poate remarca faptul că toate tabelele incluse în aceste fragmentări respectă cele 3 reguli, deci fragmentarea este corectă.

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

Tabelele replicate din baza de date inițială sunt **BOOK** și **REVIEW.** S-a ales replicarea acestor tabele, deoarece utilizatorii aplicației, indiferent de locație (Europa sau afara Europei) trebuie să aibă acces la toate cărțile și toate recenziile acestora.

În urma fragmentarii verticale, au rezultat două fragmente situate în:

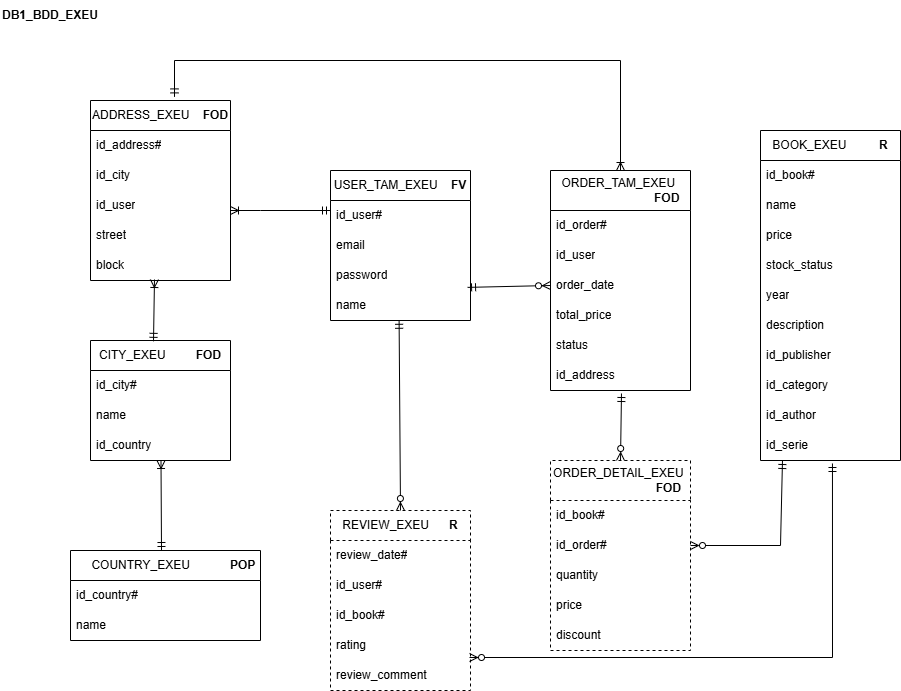
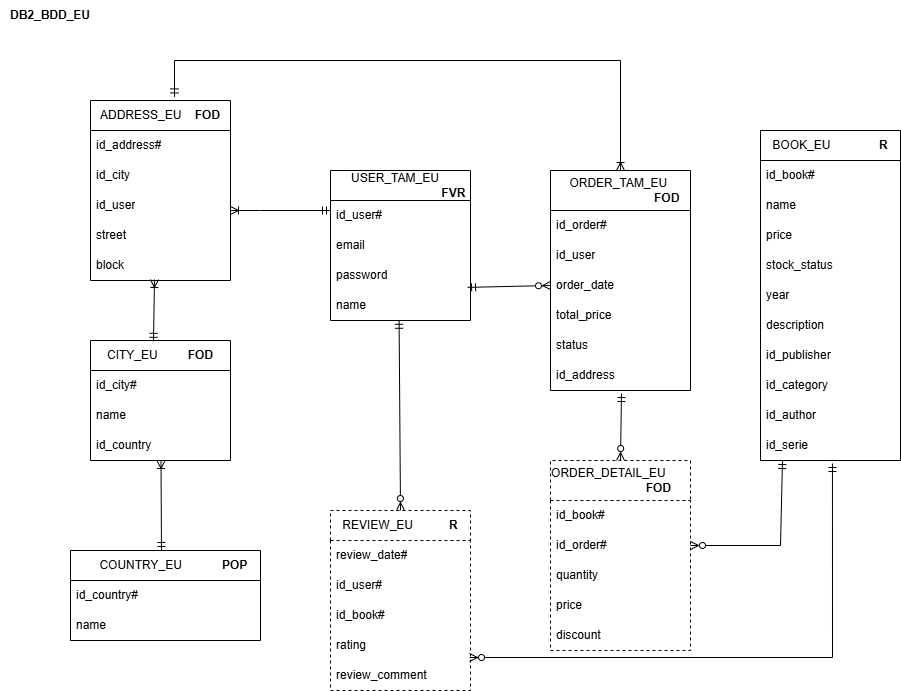
* schema utilizatorului BDD\_GLOBAL (USER\_TAM\_GLOBAL)

Fragmentul USER\_TAM\_GLOBAL a fost stocat pe stația care corespunde Europei din orcl2, deoarece datele conținute în acesta sunt sensibile și sunti utilizate în anumite cazuri speciale. Astfel datele fiind în altă bază de date sunt protejate.

* utilizatorul BDD din prima bază de date (EXEU - USER\_TAM\_EXEU)

A fost replicat fragmentul USER\_TAM\_EXEU ca **USER\_TAM\_EU** în a doua bază de date. Decizia de replicare a fost luată pentru a face posibilă conectarea și gestionarea utilizatorilor indiferent de zona geografică.

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



## 8. Lista tuturor constrângerilor ce trebuie îndeplinite de model

1. Unicitate
2. unicitate locală

Unicitate pe coloana **name** din tabelele country\_exeu și country\_eu presupune aplicarea constrângerii de unicitate la nivelul liniei din tabel.

Unicitate pe coloana **email** din tabelele user\_tam\_exeu, user\_tam\_eu presupune aplicarea constrângerii de unicitate la nivelul liniei din tabel.

1. unicitate globală pe fragmente orizontale

Se creează câte un trigger în fiecare bază de date (eu și exeu) care să verifice ca înainte de inserarea sau actualizarea numelui în tabela country\_eu (bd\_eu), respectiv country\_exeu (bd\_exeu) să nu fie în cealaltă bază de date deja un country cu acest nume deja inserat. Astfel, este asigurată unicitatea la nivel global:

- dacă name este din Europa și dacă name nu se găsește în ExEu se inserează în Eu;

- daca name este din ExEu și dacă name nu se găsește în Eu se inserează în ExEu.

1. 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 - **Nu există cazul, am vorbit la consultații**

b. PK (local, global)

În cazul fragmentării verticale constrângerea de cheie primară se implementează prin definirea constrângerii la nivelul fiecărui fragment vertical.

Se creează cheie primară pentru tabelele rezultate din fragmentarea verticală:

* user\_tam\_exeu (bd\_exeu), user\_tam\_global (schema bdd\_global din bd\_exeu), user\_tam\_eu (bd\_eu) adaugă constrângerea de cheie primară pentru coloana id\_user.

Această metodă asigură păstrarea constrângerii de cheie primară în cazul global, deoarece în momentul inserării se inserează în același timp cu aceeași valoare pentru id\_user în toate cele 3 tabelele.

În cazul fragmentării orizontale constrângerea de cheie primară se va implementa prin: definirea constrângerii la nivelul fiecărui fragment orizontal și utilizarea obiectelor de tip secvență care vor genera valori unice la nivel global. Se fac pentru fiecare tabel fragmentat orizontal câte două secvențe pe local, una pentru generare de id-uri pare (baza de date EXEU), alta pentru impare ( baza de date EU).

Astfel, cazul inserării în view-uri pe global se vor utiliza cele două secvențe de pe local pentru fragmentele respectiv, ceea ce asigură că se respectă constrângerea de cheie primară.

Se creează cheie primară pentru tabelele rezultate din fragmentarea orizontală:

* country\_eu - cu secvența de numere impare, country\_exeu - cu secvența de numere pare, id\_country va fi setat pentru fiecare fragment să aibă ca valoare default valoarea urmatoare (nextval) din secvența corespunzătoare
* city\_eu - cu secvența de numere impare, city\_exeu - cu secvența de numere pare, id\_city va fi setat pentru fiecare fragment să aibă ca valoare default valoarea urmatoare (nextval) din secvența corespunzătoare
* address\_eu - cu secvența de numere impare, address\_exeu- cu secvența de numere pare, id\_address va fi setat pentru fiecare fragment să aibă ca valoare default valoarea urmatoare (nextval) din secvența corespunzătoare
* order\_tam\_eu- cu secvența de numere impare, order\_tam\_exeu- cu secvența de numere pare, id\_order va fi setat pentru fiecare fragment să aibă ca valoare default valoarea urmatoare (nextval) din secvența corespunzătoare
* order\_detail\_eu , order\_detail\_exeu, cheia primară este compusă din id\_order și id\_book. Pentru a păstra constrângerea de cheie primară în global, se creează doi triggeri în cele două baze de date care înainte de inserarea în tabelele order\_detail\_eu, respectiv order\_detail\_exeu verifică dacă este deja inserată acea combinație de id\_order și id\_book în cealaltă bază de date, iar dacă nu este, atunci se poate insera.

De asemenea, se creează chei primare și pentru tabelele replicate

Book\_eu - cheie primară pe id\_book

Book\_exeu - cheie primară pe id\_book și crearea unei secvențe care crește cu o unitate, iar id\_book are setata ca valoare default valoarea următoare din secventă.

Constrângerea de cheie primară pentru Book în global este asigurată, deoarece, fiind o singură secvență aceasta va fi folosită când se inserează în Book\_exeu, iar prin replicare se activează triggerul care după ce se realizează inserarea în book\_exeu, aceleași date sunt inserate și în Book\_eu.

Review\_eu, Review\_exeu - cheie primară compusă cu coloanele: review\_date, id\_user, id\_book.

Constrângerea de cheie primară pentru Review în global este asigurată, deoarece, combinația este unică, iar după ce se inserează în Review\_exeu, se activează triggerul prin care se inserează în Review\_eu aceleași date care au fost introduse în Review\_exeu.

c. FK (local, pentru relații stocate în baze de date diferite)

**Nu există situația de FK pentru relații stocate în baze de date diferite, am vorbit la consultații.**

Se creează chei externe între tabelele din aceeași bază de date, anume: chei externe între tabelele din bd\_exeu și chei externe între tabelele din bd\_eu.

City\_eu - fk pe coloana **id\_country** din country\_eu

Address\_eu - fk pe coloana **id\_city** din city\_eu și **id\_user** din user\_tam\_eu

Order\_tam\_eu - fk pe coloana **id\_user** din user\_tam\_eu și **id\_address** din address\_eu

Order\_detail\_eu - fk pe coloana **id\_order** din order\_tam\_eu și **id\_book** din book\_eu

Review\_eu - fk pe coloana **id\_user** din user\_tam\_eu și **id\_book** din book\_eu

City\_exeu - fk pe coloana **id\_country** din country\_exeu

Address\_exeu - fk pe coloana **id\_city** din city\_exeu și **id\_user** din user\_tam\_exeu

Order\_tam\_exeu - fk pe coloana **id\_user** din user\_tam\_exeu și **id\_address** din address\_exeu

Order\_detail\_exeu - fk pe coloana **id\_order** din order\_tam\_exeu și **id\_book** din book\_exeu

Review\_exeu - fk pe coloana **id\_user** din user\_tam\_exeu și **id\_book** din book\_exeu

Constrângerile de cheie externă de la nivel local asigură constrângerea de cheie externă de la nivel global.

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

**Nu există validare pe relații stocate în baze de date diferite, am vorbit la consultații.**

În tabelul user\_tam\_global din schema userului bdd\_global din bd\_exeu trebuie implementate validările pe coloanele:

* is\_deleted IN (0,1)
* gender IN (‘M’,’F’)
* type IN ('Persoana fizica', 'Persoana juridica')

În tabelele review\_exeu, review\_eu se va implementa validare pe coloana:

- rating BETWEEN 1 AND 5

Constrângerea de validare la nivel local asigură respectarea constrângerii la nivel global.

## 

## 

## 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)**

Obtineti cele mai bine vandute carti per orașele Cluj-Napoca, Constanța, Iași, Timișoara, Craiova care au fost comandate (comanda avand status Completed sau Shipped) de utilizatori care inca sunt activi pe platforma si care au nume de sfant Ion, Maria, Mihail și Gavril și derivatele acestora.

Optimizări:

1. Transformare UNION în UNION ALL

Avantaje:

* îmbunătățește performanța prin eliminarea necesității de a elimina duplicatele

Dezavantaje:

* în cazul general un dezavantaj al utilizării union all este că rezultatul poate conține duplicate, iar acestea sunt necesare în logica aplicației

1. Transformare UNION ALL în IN

Avantaje:

* simplificarea logicii și a planului de execuție care duce la o reducere semnificativă a pașilor de execuție
* este recomandat de folosit doar când condiția este pe o singură coloană

1. Rescrierea inline a cererii

Avantaje:

* optimizatorul poate alege un plan de execuție mai eficient

Dezavantaj:

- codul SQL este mai greu de citit, mai puțin lizibil

1. Gruparea în WITH a părții din cerere care accesează date remote

Avantaje:

* codul SQL este mai ușor de parcurs, mai lizibil
* optimizatorul poate alege un plan de execuție mai eficient

1. Crearea de indecși pe cheile externe

Avantaje:

* reducerea timpului de execuție pentru operațiile de JOIN
* identificarea rapidă a datelor din tabelele din join

Dezavantaj:

- necesită spațiu în plus în baza de date

1. Creare de indecși pe câmpurile din filtrări

Avantaje:

* reducerea timpului de execuție pentru operațiile de tip WHERE prin utilizarea indexului în filtrare

Dezavantaje

* încetinirea operațiilor DML dacă sunt prezenți prea mulți indecși pe tabelul respectiv

1. Utilizare hint MERGE

Avantaje:

* integrează cererea asupra căreia se alege aplicarea Merge în planul cererii principale și permite optimizatorului găsirea unui plan de execuție mai eficient

Dezavantaje

* dacă interogarea vizualizării conține o clauză GROUP BY sau un operator DISTINCT în lista SELECT, atunci optimizatorul nu poate îmbina interogarea vizualizării în instrucțiunea de accesare dacă nu este activată îmbinarea complexă a vizualizărilor.

1. Utilizare hint DRIVING\_SITE pentru a muta execuția pe baza de date remote

Avantaje:

* permite execuția din baza de date selectată
* reducerea volumului datelor care sunt transferate între bazele de date

Dezavantaje

* se pot ignora optimizările locale - de exemplu indecsii care au fost creați pe tabelele bazei locale