MONITORIZAREA ACTIVITĂȚII UNEI AGENȚII IMOBILIARE

Proiect Baze de Date

**Ciurescu Irina Alexandra**

Grupa 142

## Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

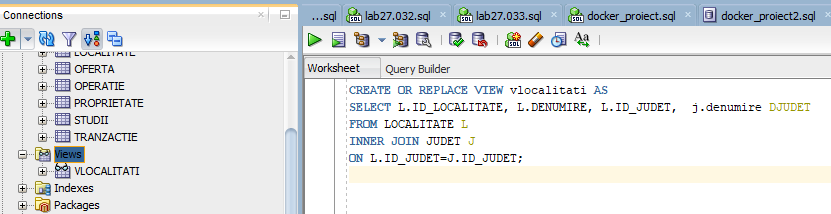
**Crearea unei vizualizări: localitățile și județele de care aparțin**.

CREATE OR REPLACE VIEW vlocalitati AS

SELECT L.ID\_LOCALITATE, L.DENUMIRE, L.ID\_JUDET, j.denumire DJUDET

FROM LOCALITATE L

INNER JOIN JUDET J

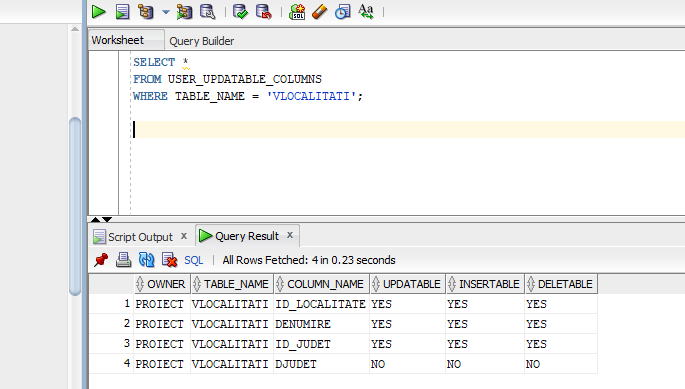
ON L.ID\_JUDET=J.ID\_JUDET;

**Coloanele updatabile din vedere**

SELECT \*

FROM USER\_UPDATABLE\_COLUMNS

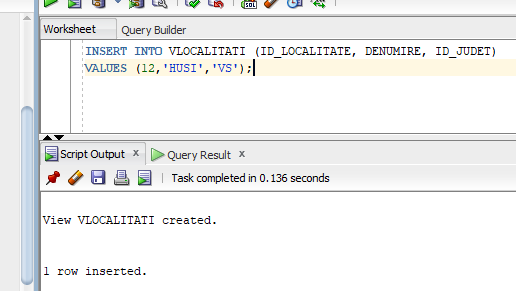
WHERE TABLE\_NAME = 'VLOCALITATI';



**Operatie LMD permisa – inserarea unei noi localități**

INSERT INTO VLOCALITATI (ID\_LOCALITATE, DENUMIRE, ID\_JUDET)

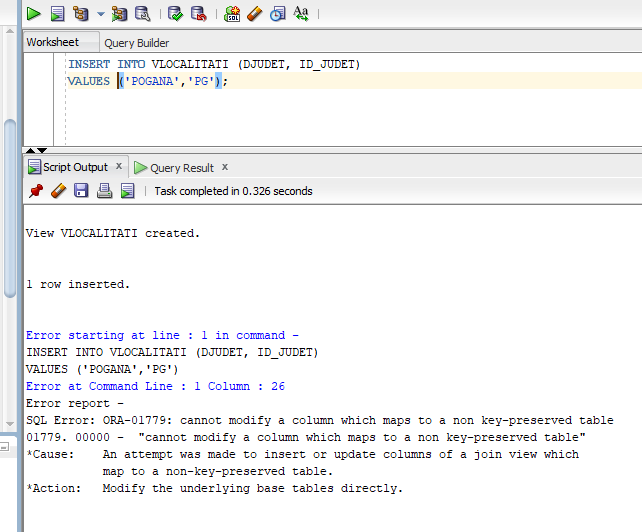
VALUES (12,'HUSI','VS');



**Operatie LMD nepermisa – inserarea unui nou județ**

INSERT INTO VLOCALITATI (DJUDET, ID\_JUDET)

VALUES ('POGANA','PG');



## Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minim 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n

1. **Pentru fiecare client să șe afișeze numărul de cereri, numărul de oferte, numărul de contracte** (cerere ce utilizează operația **outer-join** pe minimum 4 tabele)

SELECT CL1.ID\_CLIENT, CL1.NUME, CL1.PRENUME, NR\_CER, NR\_OFERTE, NR\_CONTR

FROM CLIENT CL1

LEFT OUTER JOIN

(SELECT CL.ID\_CLIENT,COUNT(CR.ID\_CLIENT) AS NR\_CER

FROM CLIENT CL

JOIN CERERE CR ON CL.ID\_CLIENT=CR.ID\_CLIENT

GROUP BY CL.ID\_CLIENT) CER ON CL1.ID\_CLIENT=CER.ID\_CLIENT

LEFT OUTER JOIN

(SELECT CL.ID\_CLIENT,COUNT(O.ID\_CLIENT) AS NR\_OFERTE

FROM CLIENT CL

JOIN OFERTA O ON CL.ID\_CLIENT=O.ID\_CLIENT

GROUP BY CL.ID\_CLIENT) O

ON CL1.ID\_CLIENT=O.ID\_CLIENT

LEFT OUTER JOIN

(SELECT CL.ID\_CLIENT, COUNT(CT.ID\_CLIENT) AS NR\_CONTR

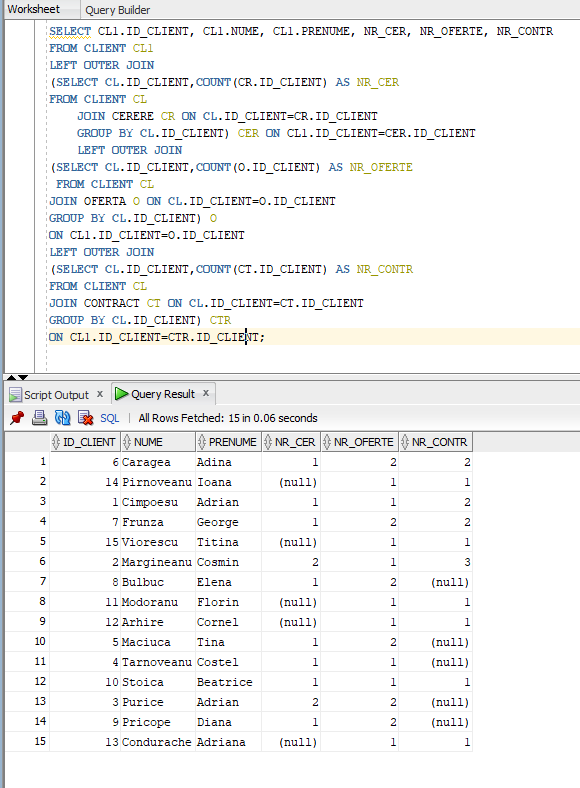
FROM CLIENT CL

JOIN CONTRACT CT ON CL.ID\_CLIENT=CT.ID\_CLIENT

GROUP BY CL.ID\_CLIENT

) CTR

ON CL1.ID\_CLIENT=CTR.ID\_CLIENT;



1. **Afișarea tuturor clienților care oferă spre vânzare sau închiriere toate cele 3 tiprui de proprietățiȘ apartamente, case, terenuri** (cerere ce utilizează operația ***division***)

SELECT ID\_CLIENT, NUME, PRENUME

FROM CLIENT

WHERE ID\_CLIENT IN

(SELECT ID\_CLIENT FROM OFERTA

MINUS

SELECT ID\_CLIENT FROM

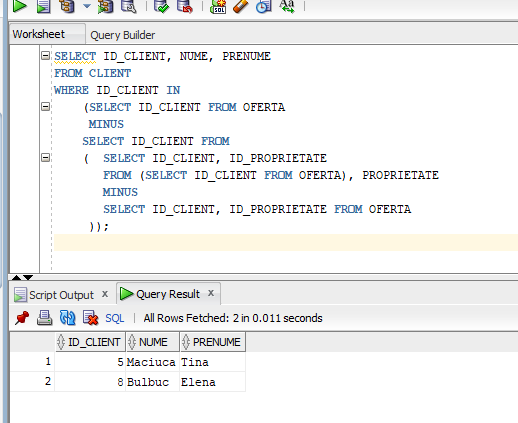
( SELECT ID\_CLIENT, ID\_PROPRIETATE

FROM (SELECT ID\_CLIENT FROM OFERTA), PROPRIETATE

MINUS

SELECT ID\_CLIENT, ID\_PROPRIETATE FROM OFERTA

));



1. **Afișati primii 2 angajați care au încheiat cele mai multe** **tranzacții**i (cerere care implementează analiza top-n).

SELECT \*

FROM (SELECT A.NUME, A.PRENUME, F.DENUMIRE FUNCTIA, TR.NR\_TRANZ

FROM ANGAJAT A

JOIN FUNCTIE F ON A.ID\_FUNCTIE = F.ID\_FUNCTIE

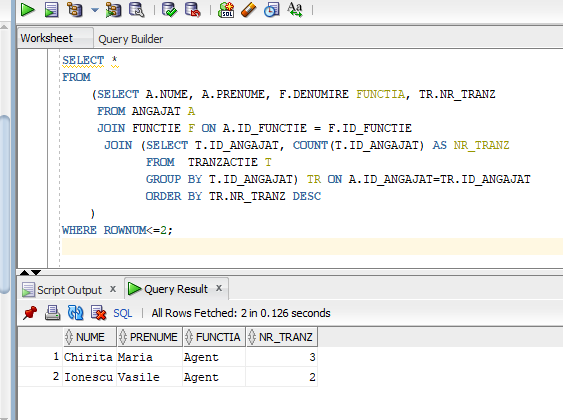
JOIN (SELECT T.ID\_ANGAJAT, COUNT(T.ID\_ANGAJAT) AS NR\_TRANZ

FROM TRANZACTIE T

GROUP BY T.ID\_ANGAJAT ) TR ON A.ID\_ANGAJAT=TR.ID\_ANGAJAT

ORDER BY TR.NR\_TRANZ DESC)

WHERE ROWNUM<=2;



## Două instrucțiuni select echivalente semantic, de comparat din punct de vedere a execuției (explicat plan de execuție).

**Afisati clienții care au tranzacționat in 2023 apartamente din București cu suprafață mai mare ca 150 mp**

**Varianta 1**

SELECT C.NUME,C.PRENUME

FROM OFERTA O

JOIN CLIENT C ON O.ID\_CLIENT=C.ID\_CLIENT

WHERE O.ID\_PROPRIETATE='A' AND O.SUPRAFATA >150

AND O.ID\_LOCALITATE IN

( SELECT L.ID\_LOCALITATE

FROM LOCALITATE L

WHERE UPPER(L.DENUMIRE) LIKE 'BUCURESTI'

)

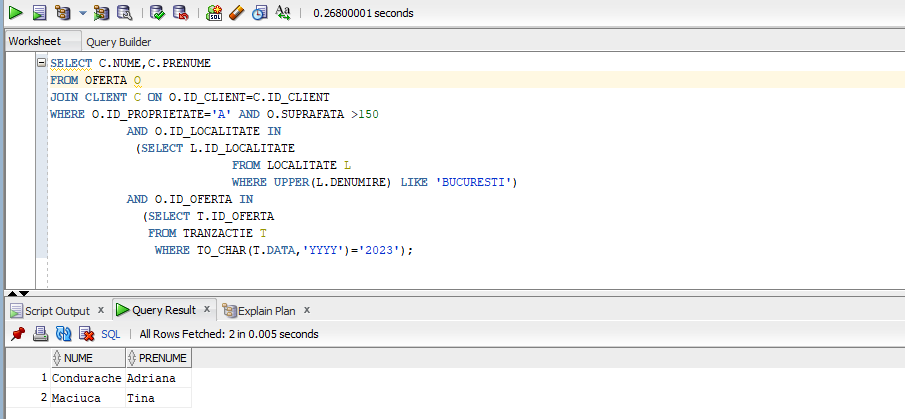
AND O.ID\_OFERTA IN

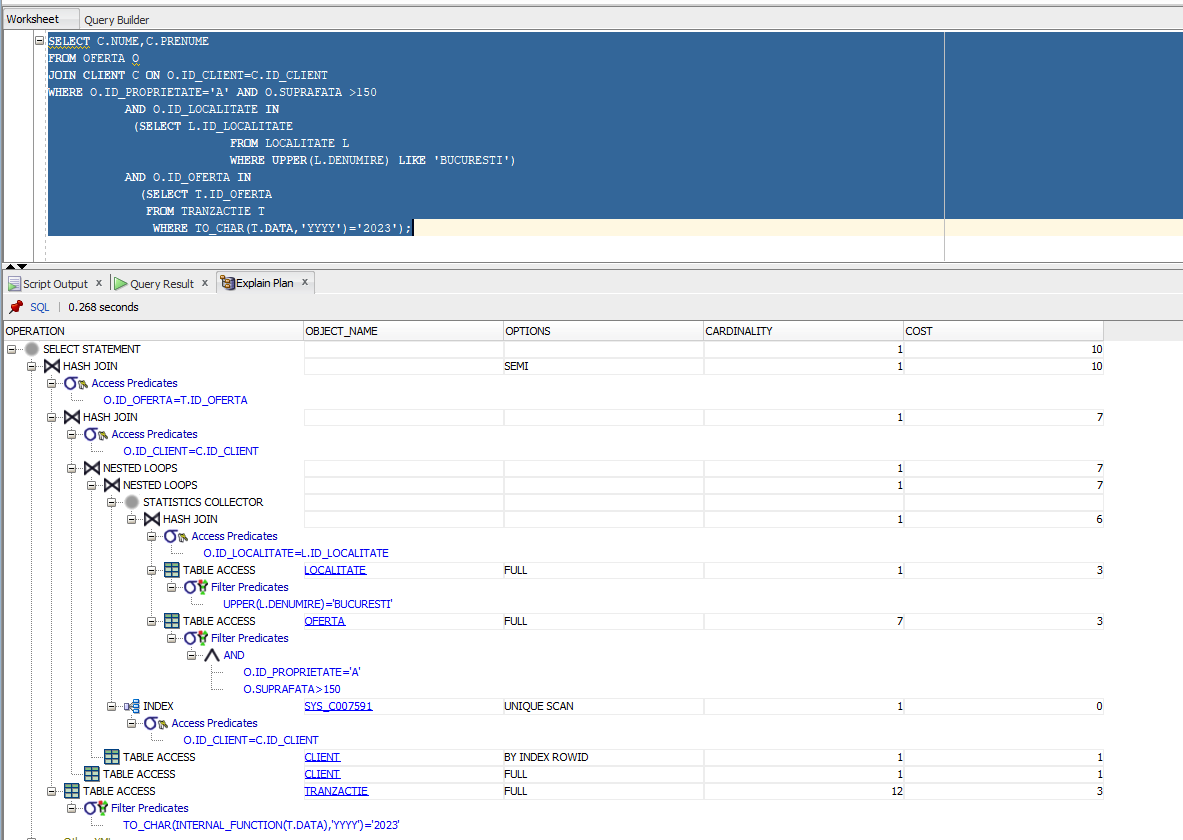
( SELECT T.ID\_OFERTA

FROM TRANZACTIE T

WHERE TO\_CHAR(T.DATA,'YYYY')='2023'

);





**Varianta 2**

WITH BUC AS

( SELECT L.ID\_LOCALITATE

FROM LOCALITATE L

WHERE UPPER(L.DENUMIRE) LIKE 'BUCURESTI'

),

DAT AS

(SELECT T.ID\_OFERTA

FROM TRANZACTIE T

WHERE TO\_CHAR(T.DATA,'YYYY')='2023'

)

SELECT C.NUME,C.PRENUME

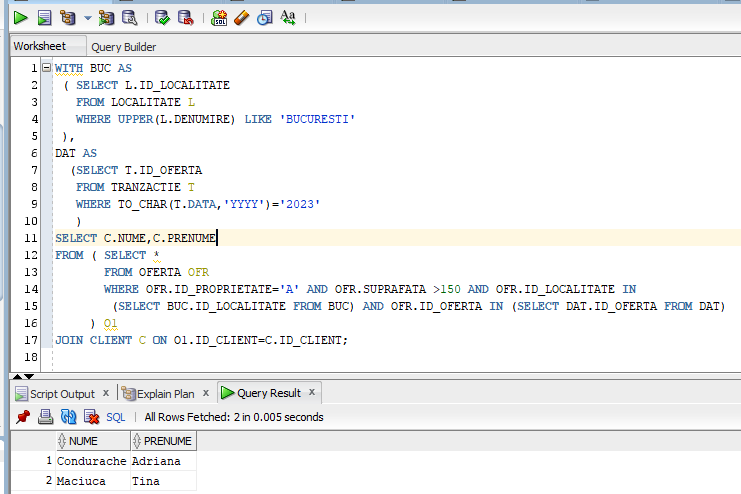
FROM ( SELECT \*

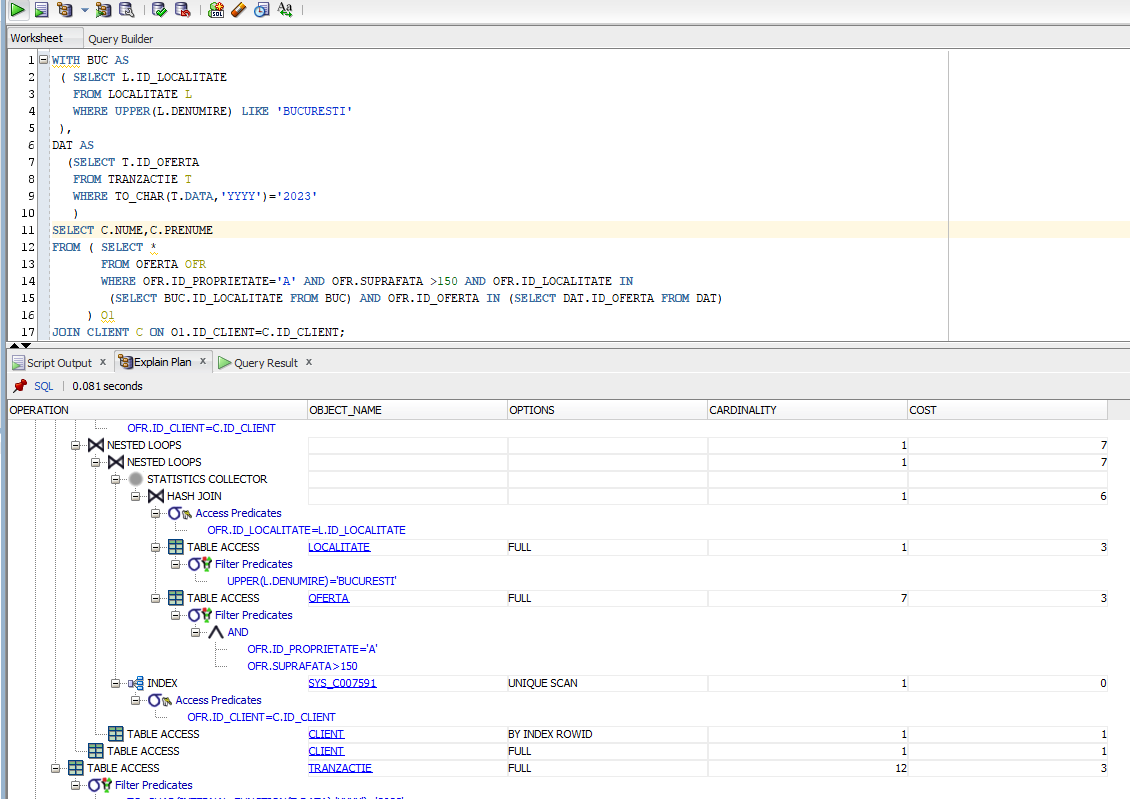
FROM OFERTA OFR

WHERE OFR.ID\_PROPRIETATE='A' AND OFR.SUPRAFATA >150 AND OFR.ID\_LOCALITATE IN

(SELECT BUC.ID\_LOCALITATE FROM BUC) AND OFR.ID\_OFERTA IN (SELECT DAT.ID\_OFERTA FROM DAT)

) O1

JOIN CLIENT C ON O1.ID\_CLIENT=C.ID\_CLIENT;



Observăm că pentru cele 2 cereri echivalente semantic planul de execuție coincide.

Planul de execuție în Oracle nu este determinat exclusiv de sintaxa interogării, ci și de structura tabelelor, statistici și alți factori relevanți pentru optimizarea interogărilor.

Optimizatorul de interogări din Oracle poate genera același plan de execuție pentru ambele interogări. El poate identifica că au aceeași semnificație logică și pot beneficia de același plan de execuție eficient. Aceasta se datorează capacității optimizatorului de a realiza optimizări logice și fizice în funcție de cerințele specifice ale interogărilor și ale bazei de date.

## Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării, justificând necesitatea acesteia.

**Realizarea normalizării BCNF (toate dependenţele sunt date de chei)**

**CONTRACT**

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_contract | Id\_angajat | Id\_client | Comision |
| 1 | 1 | 1 | 5 |
| 2 | 2 | 2 | 2 |
| 3 | 1 | 1 | 5 |
| 4 | 3 | 2 | 3 |

**Anomalii:**

Dacă agenția ar avea comisioane diferite pentru clienți, de exemplu în funcție de numărul de contracte încheiate cu aceștia, câmpul **Comision** ar fi dependent de câmpul **Id\_client** care nu este cheie primară. Apare astfel o dependență funcțională între atribute non chei.

Transformare la forma **BCNF** (reorganizăm datele adăugând o tabelă nouă)

**CONTRACT**

|  |  |  |
| --- | --- | --- |
| Id\_contract | Id\_angajat | Id\_client |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 1 | 1 |
| 4 | 3 | 2 |

**CLIENT\_COMISION**

|  |  |
| --- | --- |
| Id\_client | Comision |
| 1 | 5 |
| 2 | 2 |
| 1 | 5 |
| 2 | 3 |

**Realizarea normalizării FN4 (nu conţine două sau mai multe dependenţe multivaloare)**

**ANGAJAT**

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_angajat | Nume | Prenume | Telefon |
| 1 | Pop | Ioana | 12345 |
| 2 | Ionescu | Mihai | 12346 |
| 3 | Antoche | Ana | 12347 |

**Anomalii:**

Dacă un angajat ar avea mai multe telefoane apare o dependență multivaloare între atributul "Nume" și atributul "Telefon". Un angajat poate avea mai multe numere de telefon, dar fiecare număr de telefon este legat de un singur angajat. De asemenea, un nume de angajat poate fi asociat cu mai multe numere de telefon, dar fiecare număr de telefon este legat de un singur nume de angajat.

Transformare la forma **FN4** (reorganizăm datele adăugând o tabelă nouă)

**ANGAJAT**

|  |  |  |
| --- | --- | --- |
| Id\_angajat | Nume | Prenume |
| 1 | Pop | Ioana |
| 2 | Ionescu | Mihai |
| 3 | Antoche | Ana |

**TELEFOANE**

|  |  |  |
| --- | --- | --- |
| Id\_telefon | Id\_angajat | Telefon |
| 1 | 1 | 12345 |
| 2 | 2 | 12346 |
| 3 | 3 | 12347 |
| 4 | 1 | 12348 |

**Realizarea normalizării FN5 (fiecare dependenţă joncţiune este generată printr-un candidat cheie al tabelei)**

Baza de date nu este in forma normală FN5. De exemplu dacă un angajat este șters din tabela Angajați, toate tranzacţiile asociate cu acel angajat vor fi pierdute din tabelele Contracte şi Tranzacţii. Dacă un client este şters din tabela Clienţi atunci se vor pierde date din tabelele Cereri, Oferte, Contracte şi Tranzacţii.

De exemplu pentru tabela **CONTRACTE** ar trebui create 2 tabele noi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id\_contract | Id\_angajat | Id\_client | Data | Comision |
| 1 | 1 | 100 | 11-03-2023 | 2 |
| 2 | 2 | 89 | 10-05-2023 | 3 |
| 3 | 1 | 12 | 13-05-2023 | 2 |

**CONTRACTE**

|  |  |  |
| --- | --- | --- |
| Id\_contract | Data | Comision |
| 1 | 11-03-2023 | 2 |
| 2 | 10-05-2023 | 3 |
| 3 | 13-05-2023 | 2 |

**CONTRACTE\_ANGAJATI**

|  |  |  |
| --- | --- | --- |
| Id\_ConAng | Id\_contract | Id\_angajat |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 1 |

**CONTRACTE\_CLIENTI**

|  |  |  |
| --- | --- | --- |
| Id\_ConClient | Id\_contract | Id\_client |
| 1 | 1 | 100 |
| 2 | 2 | 89 |
| 3 | 3 | 12 |

**Denormalizarea** reprezintă procesul de adăugare deliberată a redundanței datelor într-o bază de date relațională, pentru a îmbunătăți performanța și eficiența interogărilor și a reducerii timpului de răspuns.

În timp ce normalizarea ajută la eliminarea redundanței și a dependențelor nedorite într-o bază de date, denormalizarea poate fi folosită pentru a reduce numărul de interogări necesare pentru a extrage date din baza de date. De exemplu, prin crearea unui tabel cu date redundante, cum ar fi un tabel de raportare care combină date din mai multe tabele, se poate obține o performanță mai bună și timp de răspuns mai rapid pentru interogările frecvente.

Putem denormaliza de exemplu tabelul **OFERTE** pentru a evita unirea tabelelor când se fac interogări. Tabelul **OFERTE\_DENORM** are schema relațională:

**OFERTE\_DENORM (**Id\_oferta, Id\_client, **NumeClient, PrenumeClient**, Id\_operatie, **DenumireOper**, Id\_proprietate, **DenumireProp**, Id\_localitate, **DenumireLoc**, Strada, Numar, Suprafața, An\_construcție, PrețDescriere**).**

## Tranzacții: ilustrarea consistency levels in Oracle cu tranzacții care operează asupra modelului ales (model consistency\_level.sql)

În Oracle există nivelurile de consistență Read Committed, Serializable, Read Committed with Row-Level Locking.

* **Read Committed (Implicit)**: nivelul implicit de consistență în Oracle. În acest nivel, fiecare instrucțiune select va vedea doar datele confirmate (committed) în momentul în care instrucțiunea a început să se execute. Modificările neconfirmate (uncommitted) realizate de alte tranzacții nu vor fi vizibile.
* **Serializable**: acest nivel de consistență oferă cea mai înaltă consistență. În acest nivel, tranzacțiile sunt executate în mod izolat una față de cealaltă, asigurând că orice modificare realizată de o tranzacție nu va fi vizibilă pentru alte tranzacții până când tranzacția curentă nu este finalizată. Acest nivel asigură că nu apar fenomene precum citirea datelor murdare sau fenomenul pierderii actualizărilor.
* **Read Committed with Row-Level Locking:** acest nivel de consistență oferă un grad mai mare de izolare decât Read Committed implicit. În plus față de consistența implicită, acest nivel utilizează încuietori la nivel de rând pentru a evita fenomenul citirii nerepetabile dar permite fenomenul citirii fantomă.

Pentru a utiliza aceste niveluri de consistență în Oracle, putem utiliza clauza SET TRANSACTION în tranzacțiile SQL, specificând nivelul de consistență dorit.

**Exemplu: Reducerea comisionului cu 2% pt. clienții care au mai multe contracte**

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

UPDATE CONTRACT

SET COMISION=(COMISION-2)

WHERE COMISION>2 AND ID\_CLIENT IN

(SELECT CL.ID\_CLIENT

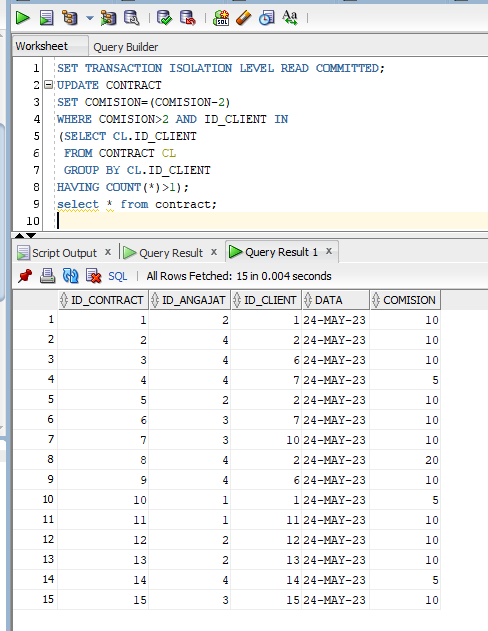
FROM CONTRACT CL

GROUP BY CL.ID\_CLIENT

HAVING COUNT(\*)>1);

SELECT \* FROM CONTRACT

COMMIT;



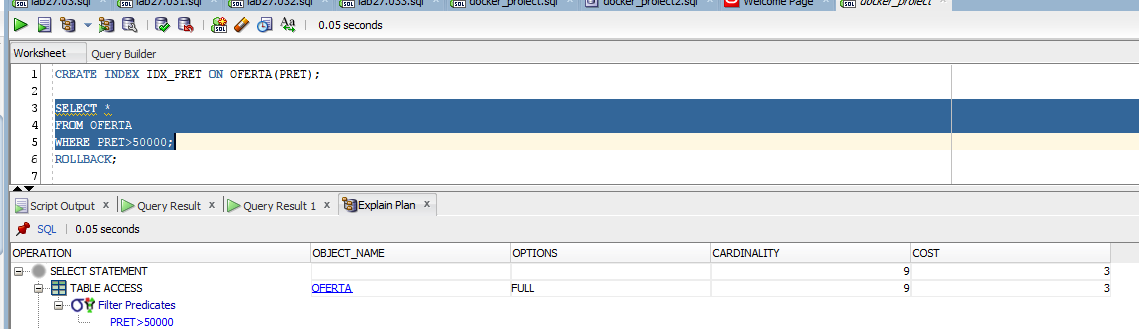
## Optimizarea a două cereri utilizând indexare

**Afișarea ofertelor pentru care prețul este mai mare decât 50.000.**

SELECT \*

FROM OFERTA

WHERE PRET>50000;



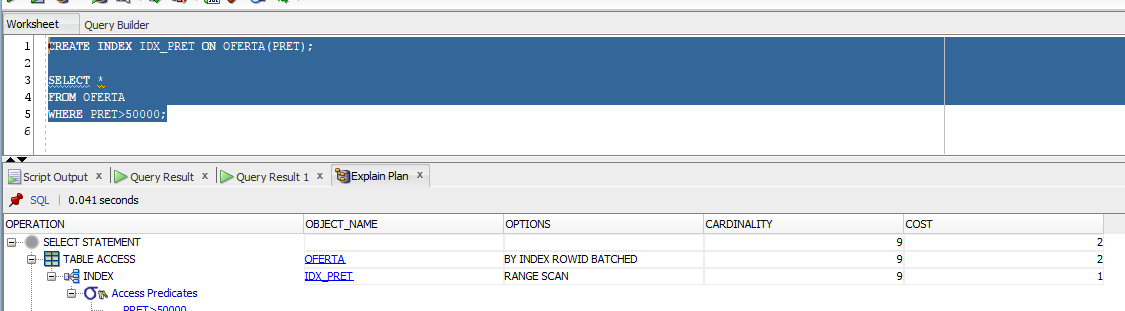
Pentru a optimiza această cerere vom crea un index pe coloana "Preț". Acest index va permite bazei de date să identifice rapid înregistrările cu prețuri mai mari decât valoarea specificată, reducând timpul de căutare.

CREATE INDEX IDX\_PRET ON OFERTA(PRET);

SELECT \*

FROM OFERTA

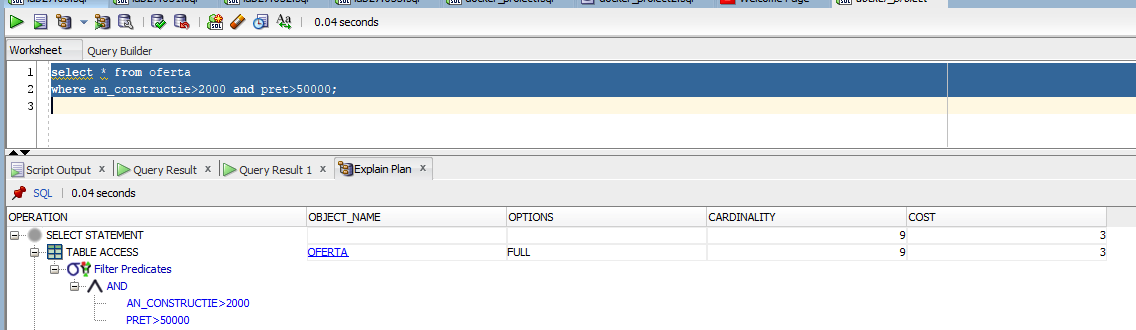
WHERE PRET>50000;



**Afișarea proprietăților construite după anul 2000 care au preț mai mare decât 50000.**

SELECT \* FROM OFERTA

WHERE AN\_CONSTRUCTIE>2000 AND PRET>50000;

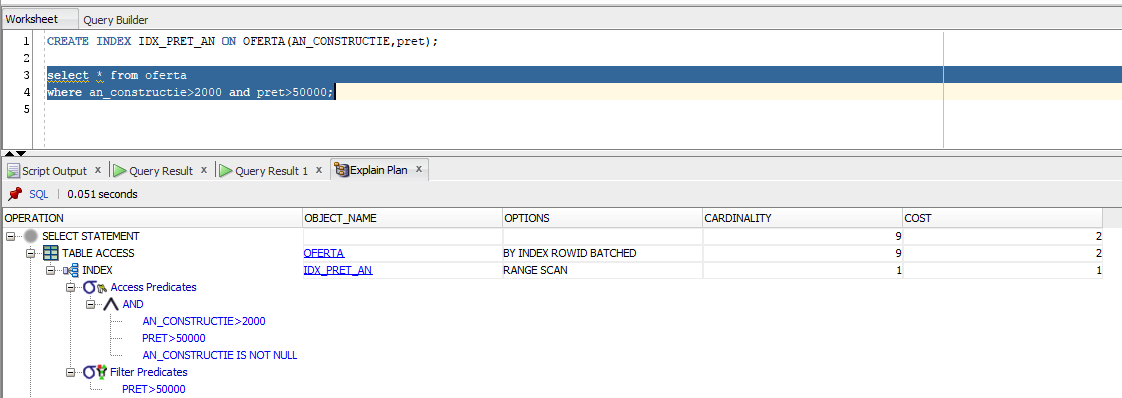


Pentru a optimiza această cerere vom crea un index compus pe coloanele "Preț" și "An construcție" . Acest index va permite bazei de date să identifice rapid înregistrările cu prețuri mai mari decât valoarea specificată construite după anul 2000, reducând timpul de căutare.

CREATE INDEX IDX\_PRET\_AN ON OFERTA(AN\_CONSTRUCTIE,PRET);

SELECT \* FROM OFERTA

WHERE AN\_CONSTRUCTIE>2000 AND PRET>50000;



Cuprins

[14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă. 1](#_Toc137380412)

[15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minim 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n 3](#_Toc137380413)

[16. Două instrucțiuni select echivalente semantic, de comparat din punct de vedere a execuției (explicat plan de execuție). 6](#_Toc137380414)

[17. Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării, justificând necesitatea acesteia. 9](#_Toc137380415)

[18. Tranzacții: ilustrarea consistency levels in Oracle cu tranzacții care operează asupra modelului ales. (model consistency\_level.sql) 12](#_Toc137380416)

[19. Optimizarea a două cereri utilizând indexare 14](#_Toc137380417)