

Universitatea din Bucureşti
Facultatea de Matematică-Informatică

PROIECT BAZE DE DATE

Voiculescu Alina-Virginia

Grupa 131

Cuprins

1. [Descriere model real.](#)
2. [Prezentare constrângeri.](#)
3. [Descriere entități.](#)
4. [Descriere relații.](#)
5. [Descriere atributelor.](#)
6. [Diagramă entitate-relație.](#)
7. [Diagramă conceptuală.](#)
8. [Scheme relaționale.](#)
9. [Normalizare \(FN1-FN3\).](#)
10. [Creare tabele SQL.](#)
11. [Cereri SQL complexe.](#)
12. [Operații de actualizare/suprimare a datelor.](#)
13. [Secvență utilizată în inserarea înregistrărilor în tabele.](#)
14. [Index care optimizează o cerere de tip căutare cu 2 criterii.](#)
15. [Cerere cu OUTER-JOIN.](#)
16. [Optimizare cerere.](#)
17. [Normalizare \(BCNF, FN4, FN5\) și Denormalizare.](#)



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

Furturile bancare sunt, din păcate, din ce în ce mai practicate în cotidian. Din această cauză, dar și datorită dorinței de a prinde răufăcătorii, este necesar a fi ținută evidență furturilor bancare. Astfel, modelul real propus se numește Gestiunea furturilor bancare. El surprinde băncile afectate, clienții acestora, martorii la jaf, hoții, sentința pe care trebuie să o deservească și unde va fi petrecută, dar și situația în care se află în momentul actual – arestați sau nu.

2

Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.

Modelul prezentat conține o serie de restricții și de reguli. Acestea vor fi prezentate în paragraful următor:

Pot exista orașe fără închisori.

Toate băncile prevăzute în model au fost jefuite.

O bancă poate să nu aibă niciun client.

Nu este obligatoriu ca un jaf să aibă martori.

Hoțul de bănci, uneori, nu este găsit în momentul interogării modelului. Asta înseamnă că acesta nu este încă arestat.

Unii hoți nu primesc sentință, deoarece aceasta se referă la închisoare. Ei sunt cei cu probleme psihice grave, fiind duși în diverse centre și spitale.

Pot exista secții de poliție în care nu lucrează niciun polițist. Acestea sunt secții vechi, desființate.



Descrierea entităților, incluzând precizarea cheii primare.

ENTITATE	CHEIE PRIMARĂ	DESCRIERE
ȚARĂ	id_țară	Entitatea reprezintă țările în care pot exista închisori.
ORAȘ	id_oraș	Entitatea reprezintă orașele în care pot exista închisori.
ÎNCHISOARE	id_închisoare	Entitatea reprezintă clădirile (închisorile) în care este/va fi petrecut timpul precizat în sentință.
SENTINȚĂ	id_sentință	Entitatea reprezintă timpul care trebuie petrecut în închisoare (sentrințele) acordat(e) hoților de către judecători.
BANCĂ	id_bancă	Entitatea reprezintă clădirile (bâncile) care au fost jefuite de-a lungul timpului.
CLIENT	id_client	Entitatea reprezintă persoanele care au realizat acțiuni în bâncile jefuite și care nu sunt angajați (clientii unei bânci jefuite).
HOȚ_DE_BĂNCI	id_hoț	Entitatea reprezintă persoana care săvârșește actul infracțional (furtul bancar).
JUDECĂTOR	id_judecător	Entitatea reprezintă persoana care are dreptul să dea hoților o sentință.
JAF	id_jaf	Entitatea reprezintă actul infracțional săvârșit de hoții de bânci.
MARTOR	id_martor	Entitatea reprezintă persoanele care au participat la jaf, dar nu sunt hoți.
POLIȚIST	id_polițist	Entitatea reprezintă persoanele care au dreptul de a aresta hoții.
SECTIE_POLITIE	id_sectie	Entitatea reprezintă clădirile (secțiile) în care lucrează polițiștii.

4

Descrierea relațiilor, incluzând precizarea cardinalității acestora.

RELATIE	DESCRIERE	CARDINALITATE
ȚARĂ – ORAȘ	Relația se numește “se află în” și face conexiunea între fiecare oraș și țara în care se află.	one-to-many
ORAȘ – ÎNCHISOARE	Relația se numește “are” și întocmește o legătură între fiecare închisoare și orașul în care se află.	one-to-many
ÎNCHISOARE – SENTINȚĂ	Relația se numește “va fi petrecută” și face conexiunea între fiecare sentință și locul în care va fi petrecută.	one-to-many
ORAȘ – BANCĂ	Relația se numește “se situează în” și întocmește o legătură între fiecare bancă și orașul în care se situează.	one-to-many
BANCĂ – CLIENT	Relația se numește “are” și face conexiunea între fiecare client al unei bănci și, desigur, banca.	one-to-many
BANCĂ – JAF	Relația se numește “face parte din” și face conexiunea între fiecare jaf și banca unde s-a petrecut.	one-to-many

MARTOR – JAF	Relația se numește “a văzut” și întocmește o legătură între fiecare martor și fiecare jaf pe care l-a vizualizat.	many-to-many
HOT_DE_BĂNCI – JAF	Relația se numește “face parte la” și face conexiunea între fiecare bancă jefuită și fiecare hoț de către care a fost jefuită.	many-to-many
HOT_DE_BĂNCI – POLIȚIST	Relația se numește “arestează” și întocmește o legătură între fiecare hoț de bănci și fiecare polițist de către care a fost arestat.	many-to-many
SECTIE POLITIE – POLIȚIST	Relația se numește “lucrează la” și face conexiunea între fiecare polițist și secția de poliție la care lucrează.	one-to-many
SENTINȚĂ – JUDECĂTOR – HOT_DE_BĂNCI	Relația se numește “primește” și se referă la sentința primită de un hoț de bănci, acordată de un judecător.	Relație de tip 3 (many-to-many-to-many)



5

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

Atributele descrise mai jos au nume simple, sugestive, pentru ca diagrama entitate-relație, diagrama conceptuală și viitoarea bază de date să fie ușor de înțeles de către cei care o vizualizează.

Entitatea ȚARĂ:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_țară	VARCHAR2(5)	NOT NULL	Da
denumire_țară	VARCHAR2(30)	NOT NULL	Nu

Entitatea ORAȘ:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id oraș	NUMBER(6)	NOT NULL	Da
denumire oraș	VARCHAR2(30)	NOT NULL	Nu

Entitatea ÎNCHISOARE:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_închisoare	NUMBER(6)	NOT NULL	Da
denumire_închisoară	VARCHAR2(30)	NOT NULL	Nu

Entitatea SENTINȚĂ:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_sentință	NUMBER(6)	NOT NULL	Da
luni_de_închisoare	NUMBER(10, 2)	NOT NULL	Nu

Entitatea BANCĂ:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_bancă	NUMBER(6)	NOT NULL	Da
denumire_bancă	VARCHAR2(30)	NOT NULL	Nu

Entitatea JUDECĂTOR:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_judecător	NUMBER(6)	NOT NULL	Da
nume_judecător	VARCHAR2(30)	NOT NULL	Nu
prenume_judecător	VARCHAR2(30)	NOT NULL	Nu
vârstă	NUMBER(6)	-	Nu
denumire_tribunal	VARCHAR2(30)	NOT NULL	Nu

Entitatea CLIENT:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_client	NUMBER(6)	NOT NULL	Da
nume_client	VARCHAR2(30)	NOT NULL	Nu
prenume_client	VARCHAR2(30)	-	Nu
vârstă_client	NUMBER(6)	-	Nu

Entitatea HOT_DE_BĂNCI:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_hot	NUMBER(6)	NOT NULL	Da
nume_hot	VARCHAR2(30)	NOT NULL	Nu
prenume_hot	VARCHAR2(30)	-	Nu
vârstă_hot	NUMBER(6)	-	Nu

Entitatea MARTOR:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_martor	NUMBER(6)	NOT NULL	Da
nume_martor	VARCHAR2(30)	NOT NULL	Nu
prenume_martor	VARCHAR2(30)	-	Nu
vârstă_martor	NUMBER(6)	-	Nu

Entitatea JAF:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_jaf	NUMBER(6)	NOT NULL	Da
dată_jaf	DATE	NOT NULL	Nu

Entitatea POLITIST:

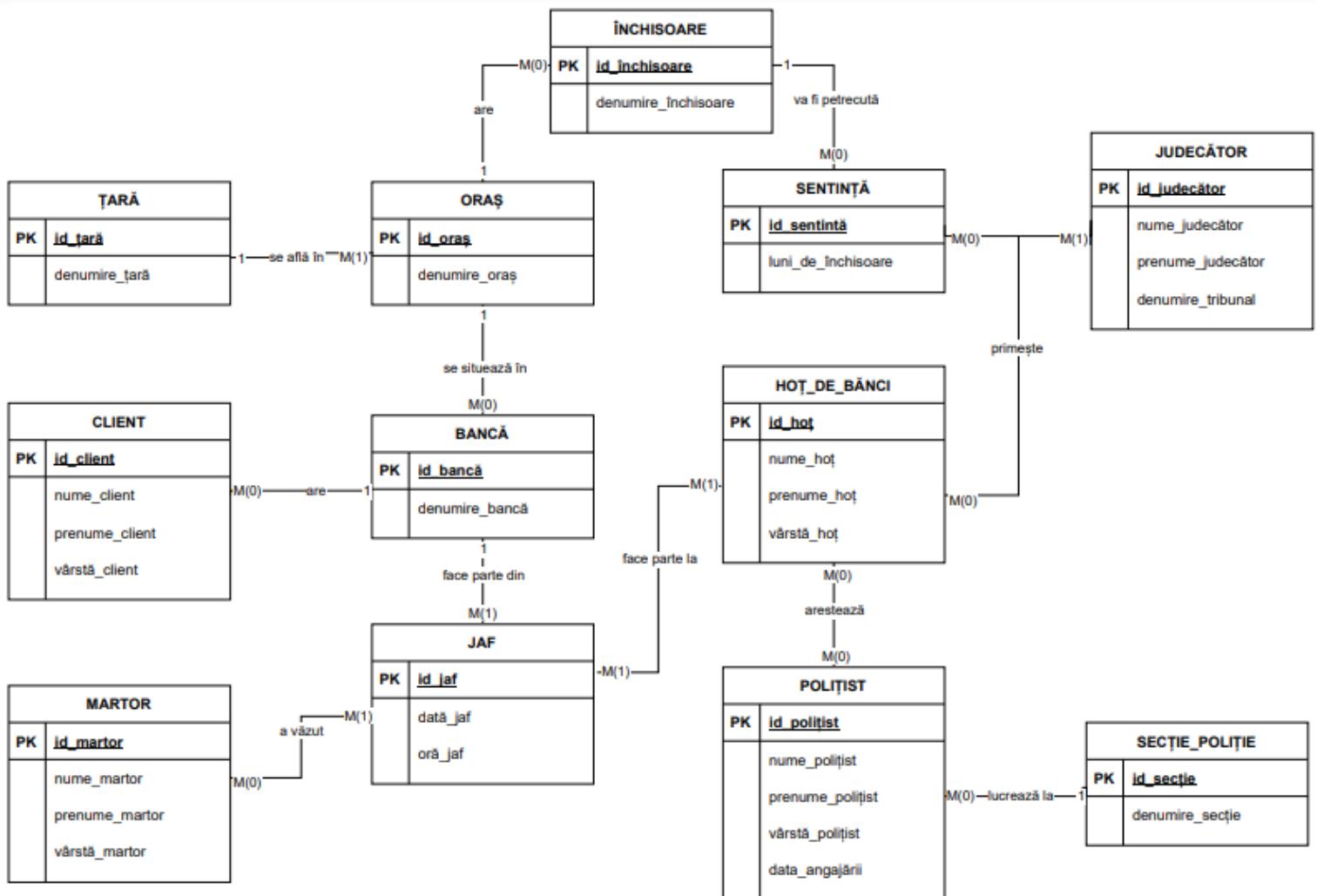
ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_politist	NUMBER(6)	NOT NULL	Da
nume_politist	VARCHAR2(30)	NOT NULL	Nu
prenume_politist	VARCHAR2(30)	-	Nu
vârstă_politist	NUMBER(6)	-	Nu
data_angajării	DATE	-	Nu

Entitatea SECTIE_POLITIE:

ATRIBUT	TIP DE DATE	CONSTRÂNGERE	CHEIE PRIMARĂ
id_sectie	NUMBER(6)	NOT NULL	Da
denumire_sectie	VARCHAR2(35)	NOT NULL	Nu

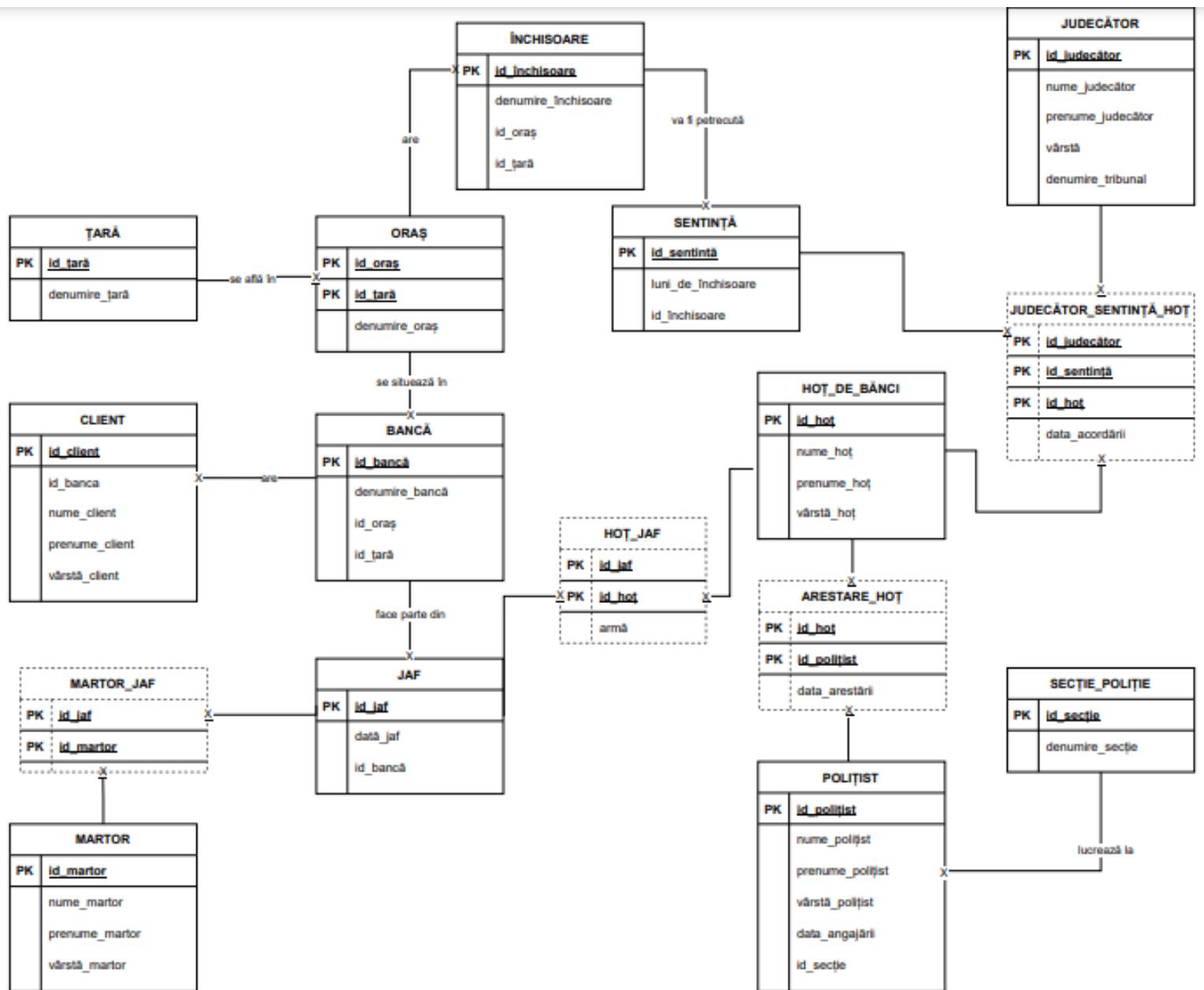
6

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





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



8

Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

ȚARĂ(id_țară#, denumire_țară)

ORAȘ(id_oraș#, id_țară#, denumire_oraș)

ÎNCHISOARE(id_închisoare#, denumire_închisoare, id_oraș, id_țară)

SENTINȚĂ(id_sentință#, luni_de_închisoare, id_închisoare)

JUDECĂTOR(id_judecător#, nume_judecător, prenume_judecător, vârstă, denumire_tribunal)

HOTEL_DE_BĂNCI(id_hotel#, nume_hotel, prenume_hotel, vârstă_hotel)

JUDECĂTOR_SENTINȚĂ_HOTEL(id_judecător#, id_sentință#, id_hotel#, data_acordării)

BANCĂ(id_bancă#, denumire_bancă, id_oraș, id_țară)

CLIENT(id_client#, id_bancă, nume_client, prenume_client, vârstă_client)

JAF(id_jaf#, dată_jaf, id_bancă)

HOTEL_JAF(id_jaf#, id_hotel#, armă)

MARTOR(id_martor#, nume_martor, prenume_martor, vârstă_martor)

MARTOR_JAF(id_jaf#, id_martor#)

SECTIE(id_secție#, denumire_secție)

POLIȚIST(id_politist#, nume_politist, prenume_politist, vârstă_politist,
data_angajării, id_secție)

ARESTARE_HOT(id_hot#, id_politist#, data_arestării)



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

Forma normală 1 (FN1)

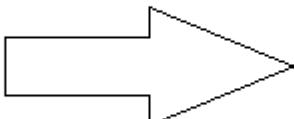
O relație este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă (atomică).

Modelul prezentat anterior este în FN1.

Exemplu NON-FN1 și transformarea în FN1:

Tabelul MARTOR_JAF:

id_jaf	id_martor
800	306
801	306
802	307, 309
803	305
804	308, 305
805	308
806	308, 309



id_jaf	id_martor
800	306
801	306
802	307
802	309
803	305
804	308
804	305
805	308
806	308
806	309

EXPLICATIE: Din cauza faptului că interogările pentru a selecta înregistrări pe baza componentei câmpurilor care conține în același timp mai multe valori semnificative sunt dificile, este nevoie ca modelul să fie în FN1. De aceea, pentru ușurință interogării datelor, în exemplul anterior a fost nevoie să trecem din non-FN1 în FN1, adăugând, pe rând, fiecare id al martorului care a participat la jaf. Pentru ca primary key-ul nostru să fie unic, el nu va mai fi format doar din id_jaf, ci din perechea (id_jaf, id_martor).

Forma normală 2 (FN2)

O relație R este în a doua formă normală dacă și numai dacă:

- Relația R este în FN1
- Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară.

Modelul prezentat anterior este în FN1, FN2.

Exemplu NON-FN2 și transformarea în FN2:

Tabelul ARESTARE_HOT:

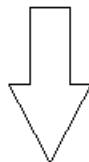
ARESTARE_HOT

id_hot	id_politist	data_arestarii	nume_hot
405	605	30-Jun-00	Osborne
405	609	20-Jul-20	Osborne
406	606	5-Apr-73	Armando
406	607	30-Dec-85	Armando
406	605	25-Apr-02	Armando
406	609	26-May-09	Armando
407	603	29-Jan-98	Derichs
408	609	27-Jan-20	Spitznagel
409	608	15-Dec-95	Jerome
409	609	15-Oct-08	Jerome



ARESTARE_HOT_1

id_hot	id_politist	data_arestarii
405	605	30-Jun-00
405	609	20-Jul-20
406	606	5-Apr-73
406	607	30-Dec-85
406	605	25-Apr-02
406	609	26-May-09
407	603	29-Jan-98
408	609	27-Jan-20
409	608	15-Dec-95
409	609	15-Oct-08



id_hot	nume_hot
405	Osborne
406	Armando
407	Derichs
408	Spitznagel
409	Jerome

ARESTARE_HOT_2

EXPLICATIE: A doua formă normală cere ca toate elementele unui tabel să fie dependente funcțional de totalitatea cheii primare. În exemplul nostru, numele hoțului depindea doar de id-ul hoțului, și nu de întreaga cheie primară, respectiv tuplul (id_hot, id_politist, data_arestarii). De aceea, a fost nevoie să transformăm acest tabel care nu este în FN2 în două tabele care sunt în FN2: primul tabel este asemănător cu cel inițial, singura diferență este că nu conține coloana nume_hot. Această coloană s-a transferat în cel de-al doilea tabel, alături de id_hot, de care depinde.

Forma normală 3 (FN3)

O relație R este în a treia formă normală dacă și numai dacă:

- Relația R este în FN2
- Fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară.

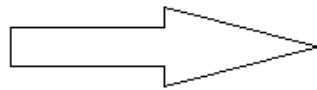
Modelul prezentat anterior este în FN1, FN2, FN3.

Exemplu NON-FN3 și transformarea în FN3:

Tabelul HOT_JAF_1:

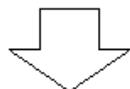
HOT_JAF_1

id_jaf	id_hot	arma	culoare_arma
800	406	pistol	gri
801	406	cutit	gri
802	409	mitraliera	negru
802	407	pusca	negru
802	405	pistol	negru
803	406	cutit	gri
804	409	electrosoc	albastru
805	406	electrosoc	galben
806	405	pistol	roz
806	408	spray paralizant	mov



HOT_JAF_11

id_jaf	id_hot	arma
800	406	pistol
801	406	cutit
802	409	mitraliera
802	407	pusca
802	405	pistol
803	406	cutit
804	409	electrosoc
805	406	electrosoc
806	405	pistol
806	408	spray paralizant



arma	culoare_arma
pistol	gri
cutit	gri
mitraliera	negru
pusca	negru
pistol	negru
cutit	gri
electrosoc	albastru
electrosoc	galben
pistol	roz
spray paralizant	mov

HOT_JAF_12

EXPLICATIE: Toate atributele care nu sunt chei ale unei relații depind numai de chei candidate ale acelei relații. Astfel, putem remarca greșeala inițială: culoarea armei depindea doar de armă, care nu face parte din cheia primară. Rezolvarea acestei probleme constă în transformarea tabelului de start în două tabele: unul care să conțină toate datele din tabelul inițial, fără coloana în care este prevăzută culoarea armei, iar celălalt – arma și culoarea acesteia.

P.S.: Aceste tabele nu există în model, fiind adaptate după tabelul existent HOT_JAF. Exercițiul anterior este doar un exemplu pentru NON-FN3 și transformarea în FN3.

10

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

Acet exercițiu se află în fișierul text Voiculescu_Alina_Virginia-creare_inserare.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

Crearea tabelelor:

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\create_insert.sql
- Menu Bar:** File Edit View Navigate Run Source Team Tools Window Help
- Toolbar:** Standard toolbar icons.
- Connections:** ProjectBD is selected.
- Connections List:** temalab5+lab6.sql, lab8.sql, temalab9.sql, temalab9.sql, temalab4.sql, ProjectBD, create_insert.sql, hr_create.sql.
- Worksheet:** The main area contains the following SQL code:

```
96    varsta_martor number(6);
97  );
98
99  create table sectie_politie(
100    id_sectie number(6) primary key,
101    denumire_sectie varchar2(30) not null
102  );
103
104 create table politist(
105    id_politist number(6) primary key,
106    nume_politist varchar2(30) not null,
107    prenume_politist varchar2(30),
108    varsta_politist number(6),
109    data_angajarii date,
110    id_sectie number(6) references sectie_politie(id_sectie)
111  );
112
113
```
- Script Output:** Shows the results of the table creations:

```
Table MARTOR created.

Table SECTIE_POLITIE created.

Table POLITIST created.
```

Inserarea datelor:

ÎNAINTE:

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the connection is named 'ProjectBD' and the file is 'create_insert.sql'. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help. The toolbar has icons for New, Open, Save, Run, Stop, Refresh, Undo, Redo, and others. The Connections panel on the left shows one connection named 'ProjectBD'. The main workspace contains a worksheet tab with the following SQL code:

```
values (3, 'ES', 'mauria');

create table inchisoare(
    id_inchisoare number(6) primary key,
    denumire_inchisoare varchar2(30) not null,
    id_oras number(6),
    id_tara varchar(5),
    foreign key (id_oras, id_tara) references oras(id_oras, id_tara)
);

select * from inchisoare;

create table sentinta(
    id_sentinta number(6) primary key,
    id_inchisoare number(6) not null,
    id_tara varchar(5),
    constraint fk_sentinta_inchisoare foreign key (id_inchisoare) references inchisoare(id_inchisoare),
    constraint fk_sentinta_tara foreign key (id_tara) references tara(id_tara)
);
```

The 'Query Result' tab shows the output of the 'select * from inchisoare;' query:

ID_INCHISOARE	DENUMIRE_INCHISOARE	ID_ORAS	ID_TARA
3	mauria		

ÎN TIMPUL INSERĂRII:

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the connection is to 'C:\Users\LMP\Desktop\PRJ BD\create_insert.sql'. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Print. The Connections panel shows a single connection named 'ProjectBD'. The main workspace is a 'Worksheet' tab where a SQL script is being run. The script contains several 'insert' statements for the 'inchisoare' table and a final 'select * from inchisoare;' statement. The output pane below shows the results of each insert operation, indicating one row was inserted for each of the four entries. The bottom status bar shows 'Task completed in 0.13 seconds'.

```
insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (125, 'Inchisoarea Osaka', 1, 'JP');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (126, 'Inchisoarea St. Gilles', 4, 'BE');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (127, 'Penitenciarul Bucuresti-Jilava', 2, 'RO');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (128, 'Inchisoarea Plotzensee', 3, 'DE');

select * from inchisoare;
```

1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.

DUPĂ INSERARE:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Print, and Database. The connections pane on the left lists 'tema_lab5+lab6.sql', 'lab8.sql', 'tema_lab9.sql', and 'create_insert.sql'. The main workspace is titled 'Worksheet' and contains the following SQL code:

```
insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (125, 'Inchisoarea Osaka', 1, 'JP');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (126, 'Inchisoarea St. Gilles', 4, 'BE');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (127, 'Penitenciarul Bucuresti-Jilava', 2, 'RO');

insert into inchisoare (id_inchisoare, denumire_inchisoare, id_oras, id_tara)
values (128, 'Inchisoarea Plotzensee', 3, 'DE');

select * from inchisoare;
```

The 'Script Output' tab at the bottom shows the results of the query:

ID_INCHISOARE	DENUMIRE_INCHISOARE	ID_ORAS	ID_TARA
1	124 Inchisoarea Aranjuez	5 ES	
2	125 Inchisoarea Osaka	1 JP	
3	126 Inchisoarea St. Gilles	4 BE	
4	127 Penitenciarul Bucuresti-Jilava	2 RO	
5	128 Inchisoarea Plotzensee	3 DE	

11

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

- operație join pe cel puțin 4 tabele
- filtrare la nivel de linii
- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în care intervin cel puțin 3 tabele
- grupări de date, funcții grup, filtrare la nivel de grupuri
- ordonări
- utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere(clauza WITH)

Acest exercițiu se află în fișierul text Voiculescu_Alina_Virginia-exemple.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

Implementarea cererilor și rularea codului:

--Exercițiu 1--

Afișați denumirea țărilor în care se află închisorile în care au mers/vor merge hoții a căror sentință a fost stabilită de judecătorul care are a treia literă din prenume 'w' și lungimea numelui egală cu 8. Ordonați-le descrescător după numele țării. Afișați și data acordării sentinței. Rezultatul va avea încă o coloană în care va fi afișată prima zi de Luni de după data acordării sentinței dacă ziua este număr par, iar dacă este număr impar, va fi afișată data corespunzătoare ultimei zile din luna din care data acordării face parte. Coloana din urmă va fi numită 'Par/Impar'. Vor fi două tipuri de soluții: decode si case.

Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

tema_lab5-Hab6.sql tema_lab9.sql tema_lab9.sql cerinte.sql

SQL Worksheet History

Worksheet Query Builder

```
--afisati denumirea tarilor in care se afla inchisorile in care au mers/vor merge hotii a caror sentinta a fost stabilita de judecatorul care are a treia litera din prenume 'w'
--si lungimea numelui egala cu 8. Ordonati-le descrescator dupa numele tarii. Afisati si data acordarii sentintei. Rezultatul va avea inca o coloana in care va fi afisata
--prima zi de Luni de dupa data acordarii sentintei daca ziua este numar par, iar daca este numar impar, va fi afisata data corespunzatoare ultimei zile din luna din care
--data acordarii face parte. Coloana din urma va fi numita 'Par/Impar'. Vor fi doua tipuri de solutii: decode si case.

--decode
select denumire_tara, data_acordarii,
decode(mod(to_char(data_acordarii, 'DD'), 2),
0, to_char(next_day(data_acordarii, 'Monday')),
to_char(last_day(data_acordarii))) "Par/Impar"
from tara
join oras using(id_tara)
join inchisoare using(id_oras)
join sentinta using(id_inchisoare)
join judecator_sentinta_hot using(id_sentinta)
join judecator using(id_judecator)
where substr(prenume_judecator, 3, 1) = 'w'
and length(nume_judecator) = 8
order by 1 desc;
```

Query Result

	DENUMIRE_TARA	DATA_ACORDARII	Par/Impar
1	Romania	20-JAN-20	27-JAN-20
2	Germania	12-OCT-08	13-OCT-08
3	Belgia	25-MAY-09	31-MAY-09
4	Belgia	17-JUL-20	31-JUL-20

Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

tema_lab5-Hab6.sql tema_lab9.sql tema_lab9.sql cerinte.sql

SQL Worksheet History

Worksheet Query Builder

```
order by 1 desc;

--case
select denumire_tara, data_acordarii,
case mod(to_char(data_acordarii, 'DD'), 2)
    when 0 then to_char(next_day(data_acordarii, 'Monday'))
    else to_char(last_day(data_acordarii))
end "Par/Impar"
from tara
join oras using(id_tara)
join inchisoare using(id_oras)
join sentinta using(id_inchisoare)
join judecator_sentinta_hot using(id_sentinta)
join judecator using(id_judecator)
where substr(prenume_judecator, 3, 1) = 'w'
and length(nume_judecator) = 8
order by 1 desc;
--2
```

Query Result

	DENUMIRE_TARA	DATA_ACORDARII	Par/Impar
1	Romania	20-JAN-20	27-JAN-20
2	Germania	12-OCT-08	13-OCT-08
3	Belgia	25-MAY-09	31-MAY-09
4	Belgia	17-JUL-20	31-JUL-20

--Exercițiul 2--

Afișați id-ul fiecărui jaf care a avut mai mult de un martor, numărul de martori, cât și numele celui mai mic martor care a asistat la jaf. Ultima coloană se va numi Mezin.

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the path is C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for temalab5+lab6.sql, temalab9.sql, temalab9.sql, and cerinte.sql. The cerinte.sql tab is active, displaying the following SQL code:

```
join judecator using(id_judecator)
where substr(prenume_judecator, 3, 1) = 'w'
and length(nume_judecator) = 8
order by 1 desc;

--2
--Afisati id-ul fiecarui jaf care a avut mai mult de un martor, numarul de martori, cat si numele celui mai mic martor care a asistat la jaf. Ultima coloana se va numi Mezin.

with martor_age as (select id_martor, nvl(varsta_martor, 1000) as AGE
                     from martor)
select id_jaf, count(id_martor) "Nr. martori", (select nume_martor
                                                   from martor
                                                 where varsta_martor = (select min(age)
                                                               from martor_age
                                                               join martor_jaf using (id_martor)
                                                               where id_jaf = mj.id_jaf)) "Mezin"
from martor_jaf mj
group by id_jaf
having count(*) > 1;
```

In the bottom pane, the Query Result tab shows the execution results:

ID_JAF	Nr. martori	Mezin
1	802	2 Mendez
2	804	2 Norman
3	806	2 Wu

--Exercițiul 3--

Afișați numele și prenumele polițistului care a arestat cei mai mulți răușăcători, alături de primul hoț de bănci prins de acesta.

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The title bar indicates the current file is 'cerinte.sql'. The main window has tabs for 'tema_lab5+Lab6.sql', 'tema_lab9.sql', 'tema_lab9.sql', and 'cerinte.sql'. The 'Worksheet' tab is active, displaying the following complex SQL query:

```
with prim_hot as (select id_politist, min(data_arestarii) as Dataarest
                   from arestare_hot
                   group by id_politist),
politist_1 as (select id_politist, nume_politist, prenume_politist
                  from politist
                  join arestare_hot using (id_politist)
                  group by id_politist, nume_politist, prenume_politist
                  having count(*) = (select max(count(*))
                                      from politist
                                      join arestare_hot using(id_politist)
                                      group by id_politist))
select nume_politist, prenume_politist, nume_hot || ' ' || prenume_hot as "Nume hot"
from politist_1 p
join arestare_hot a on (a.id_politist = p.id_politist)
join hot_de_banci h on (a.id_hot = h.id_hot)
where h.id_hot = (select id_hot
                   from arestare_hot
                   where data_arestarii = (select Dataarest
                                             from prim_hot
                                             where id_politist = p.id_politist))
group by nume_politist, prenume_politist, nume_hot || ' ' || prenume_hot;
```

The 'Query Result' tab at the bottom shows the output of the query:

NUME_POLITIST	PRENUME_POLITIST	Nume hot
1 Voiculescu	Alina	Jerome Veronika

All Rows Fetched: 1 in 0.221 seconds

--Exercițiul 4--

Găsiți cuplul martor la unul dintre jafuri și afișați numele de familie al acestora.

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar below has icons for New, Open, Save, Run, Stop, Refresh, Undo, Redo, Copy, Paste, Find, and Replace. The Connections panel on the left lists 'tema_lab5+lab6.sql', 'tema_lab9.sql', 'tema_lab9.sql', and 'cerinte.sql'. The main area has tabs for Worksheet and Query Builder, with 'Worksheet' selected. The code in the Worksheet tab is:

```
--1
select nume_hot
from arestare_hot
where data_arestarii = (select DataAreast
                        from prim_hot
                        where id_politist = p.id_politist))
group by nume_politist, prenume_politist, nume_hot || ' ' || prenume_hot;

--2
--Găsiți cuplul martor la unul dintre jafuri și afișați numele de familie al acestora.
--3
select distinct nume_martor
from martor m
where nume_martor in (select nume_martor
                        from martor
                        where m.prenume_martor != prenume_martor);

--4
```

The Query Result tab at the bottom shows the output of the last query:

NUME_MARTOR
1 Mendez

All Rows Fetched: 1 in 0.017 seconds

--Exercițiul 5--

Afișați numele fiecărui hot, alături de denumirea băncii pe care a jefuit-o și de arma cu care a amenințat populația participantă la acțiune.

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for temă_lab5+Lab6.sql, temă_lab9.sql, temă_lab9.sql, cerinte.sql, and Voiculescu_Alina_Virginia-creare_inserare.txt. The cerinte.sql tab is active. The code area contains the following SQL query:

```
from martor
where m.prenume_martor != prenume_martor;

--5
--Afisati numele fiecarui hot, alaturi de denumirea bancii pe care a jefuit-o si de arma cu care a amenintat populatia participanta la actiune.

select distinct nume_hot || ' ' || prenume_hot as "Nume hot", denumire_banca, arma
from hot_de_banci
join hot_jaf using(id_hot)
join jaf using (id_jaf)
join banca using (id_banca);
```

The results are displayed in the 'Query Result' tab, showing 9 rows of data:

Nume hot	DENUMIRE_BANCA	ARMA
1 Armando Anika	Banque CPH	cutit
2 Osborne Kizzy	Commerzbank	pistol
3 Armando Anika	Caixabank	cutit
4 Jerome Veronika	Commerzbank	mitraliera
5 Armando Anika	Banque CPH	electroscoc
6 Jerome Veronika	BCR	electrosoc
7 Derichs Roelof	Commerzbank	pusca
8 Spitznagel Tania	Commerzbank	spray paralizant
9 Armando Anika	Mizuho Bank	pistol

12

Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.

Acest exercițiu se află în fișierul text Voiculescu_Alina_Virginia-exemple.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar below has various icons for file operations. The main workspace shows a list of open files: tema_lab5+lab6.sql, lab8.sql, tema_lab9.sql, tema_lab9.sql, create_insert.sql, INCHISOARE, and cerinte.sql. The 'cerinte.sql' tab is active. The code in the editor is:

```
133    from hot_de_panci
134    where nume_hot = 'Spitznagel'
135    and prenume_hot = 'Tania';
136
137 rollback;
138
139 --DELETE
140 --Hmm..cineva a introdus in tabelul martor o persoană care nu a fost martoră la niciun jaf. Ștergeți-o!
141
142 delete from martor
143 where id_martor not in (select distinct id_martor
144                           from martor_jaf);
145
146 rollback;
147
148
```

The 'Script Output' tab at the bottom shows the results of the execution:

```
Rollback complete.

1 row deleted.

Rollback complete.
```

13

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

Acest exercițiu se află în fișierul text Voiculescu_Alina_Virginia-exemple.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

tema_lab5+lab6.sql lab8.sql tema_lab9.sql tema_lab9.sql create_insert.sql INCHISOARE cerinte.sql

SQL Worksheet History

Worksheet Query Builder

```
149
150 -----13-----
151
152 --Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).
153
154 create sequence seq_politist
155   start with 610
156   increment by 1
157   nocycle
158   nocache;
159
160 --Exemplu de inserare date cu ajutorul secvenței create
161 insert into politist (id_politist, id_sectie, nume_politist, prenume_politist, varsta_politist, data_angajarii)
162 values (seq_politist.nextval, 600, 'LastNameSeq', 'FirstNameSeq', 100, to_date(sysdate, 'DD-MON-YY'));
163
164 rollback;
```

Script Output | Query Result | Task completed in 0.083 seconds

Sequence SEQ_POLITIST created.

1 row inserted.

Rollback complete.

14

Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.

Acum exercițiu se află în fișierul text Voiculescu_Alina_Virginia-exemple.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

Crearea indexului

Oracle SQL Developer : C:\Users\LMPC\AppData\Roaming\SQL Developer\cerinte.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

tema_lab5+lab6.sql temा_lab9.sql tema_lab9.sql Welcome Page cerinte.sql

SQL Worksheet History

Worksheet Query Builder

```
--Crearea unui index care sa optimizeze o cerere de tip cautare cu 2 criterii. Specificati cererea
--CERERE: Afisati id-ul politistului care a arestat hotul cu id-ul 409 intr-o zi de 15.

create index idx_arestare_hot on
arestare_hot(id_hot, id_politist, data_arestariei);

select id_politist
from arestare_hot
where id_hot = 409
and data_arestariei like '15-%';

-----
```

Script Output x | Query Result x

| Task completed in 0.713 seconds

Index IDXARESTARE_HOT created.

Rezultat

The screenshot shows the Oracle SQL Developer interface. The top bar displays the path "Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for "tema_lab5+lab6.sql", "tema_lab9.sql", "tema_lab9.sql", "Welcome Page", and "cerinte.sql". The "cerinte.sql" tab is active. The "Worksheet" tab is selected in the toolbar. The code area contains the following SQL:

```
--Crearea unui index care sa optimizeze o cerere de tip cautare cu 2 criterii. Specificati cererea  
--CERERE: Afisati id-ul politistului care a arestat hotul cu id-ul 409 intr-o zi de 15.  
  
create index idx_arestare_hot on  
arestare_hot(id_hot, id_politist, data_arestariei);  
  
select id_politist  
from arestare_hot  
where id_hot = 409  
and data_arestariei like '15-%';
```

The "Query Result" tab is active at the bottom, showing the output of the query:

ID_POLITIST
1
608
2
609

15

Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division.

Acest exercițiu se află în fișierul text Voiculescu_Alina_Virginia-exemple.txt. Aici vor fi doar niște print-screen-uri care pot dovedi faptul că proiectul nu a fost plagiat și a fost rulat în SQL Oracle.

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\LMP\AppData\Roaming\SQL Developer\cerinte.sql". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help. The toolbar has various icons for file operations. The Connections sidebar shows "ProjectBD". The main window has tabs for "tema_lab5+lab6.sql", "lab8.sql", "tema_lab9.sql", "tema_lab9.sql", and "cerinte.sql". The Worksheet pane contains the following SQL code:

```
anu id_martor = sus;
-----DIVISION
--Sa se obtina id-urile politistilor care i-au arestat de-alungul vremii pe fiecare dintre hotii cu id-urile 405 si 406.
select distinct id_polist
from arestare_hot
where id_hot in (select distinct id_hot
                  from hot_de_banci
                  where id_hot in (405, 406))
group by id_polist
having count(*) = (select count(*)
                      from hot_de_banci
                      where id_hot in (405, 406));
```

The Query Result pane shows the output of the query:

ID_POLIST	COUNT(*)
1	605
2	609

16

Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.

Cererea în limbaj natural:

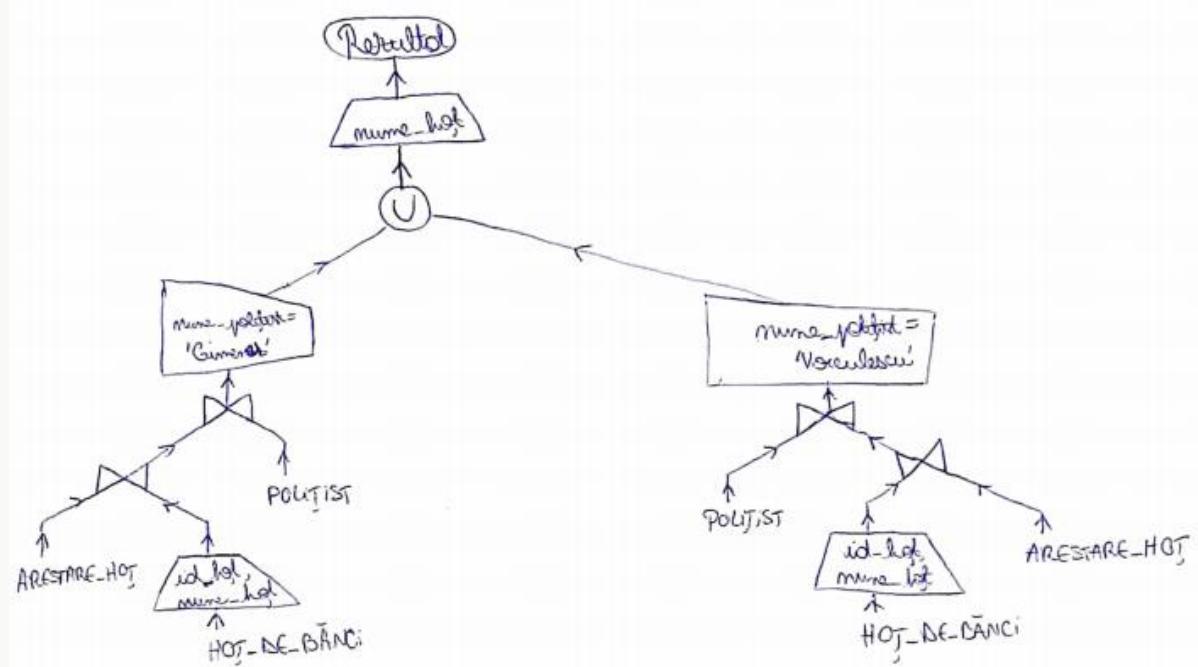
Afișați numele hoților care au fost arestați de polițiștii cu numele Gimenez și Voiculescu.

ÎNAINTE DE OPTIMIZARE

Expresie algebraică:

Rezultat = $\Pi_{\text{mure_hot}} (\nabla_{\text{mure_polist}} = \text{'Gimnast'} (\text{JOIN}(\text{JOIN}(\nabla_{\text{id_hot}}, \text{mure_hot}), \text{HOT_DE_BĂNCI}), \text{ARESTARE_HOT}), \text{POLITIST})) \cdot \nabla_{\text{mure_polist}} = \text{'Voculaseu'} (\text{JOIN}(\text{JOIN}(\nabla_{\text{id_hot}}, \text{mure_hot}), \text{HOT_DE_BĂNCI}), \text{ARESTARE_HOT}), \text{POLITIST}))$

Arbore algebraic:



Cerere în SQL:

```
| with Join1 as
|   (select id_hot, nume_hot
|    from hot_de_banci)
| select nume_hot
|   from Join1
|   join arestare_hot using(id_hot)
|   join politist using(id_politist)
| where nume_politist = 'Gimenez'
| UNION
| select nume_hot
|   from Join1
|   join arestare_hot using(id_hot)
|   join politist using(id_politist)
| where nume_politist = 'Voiculescu';
```

Query Result | All Rows Fetched: 4 in 0.004 seconds

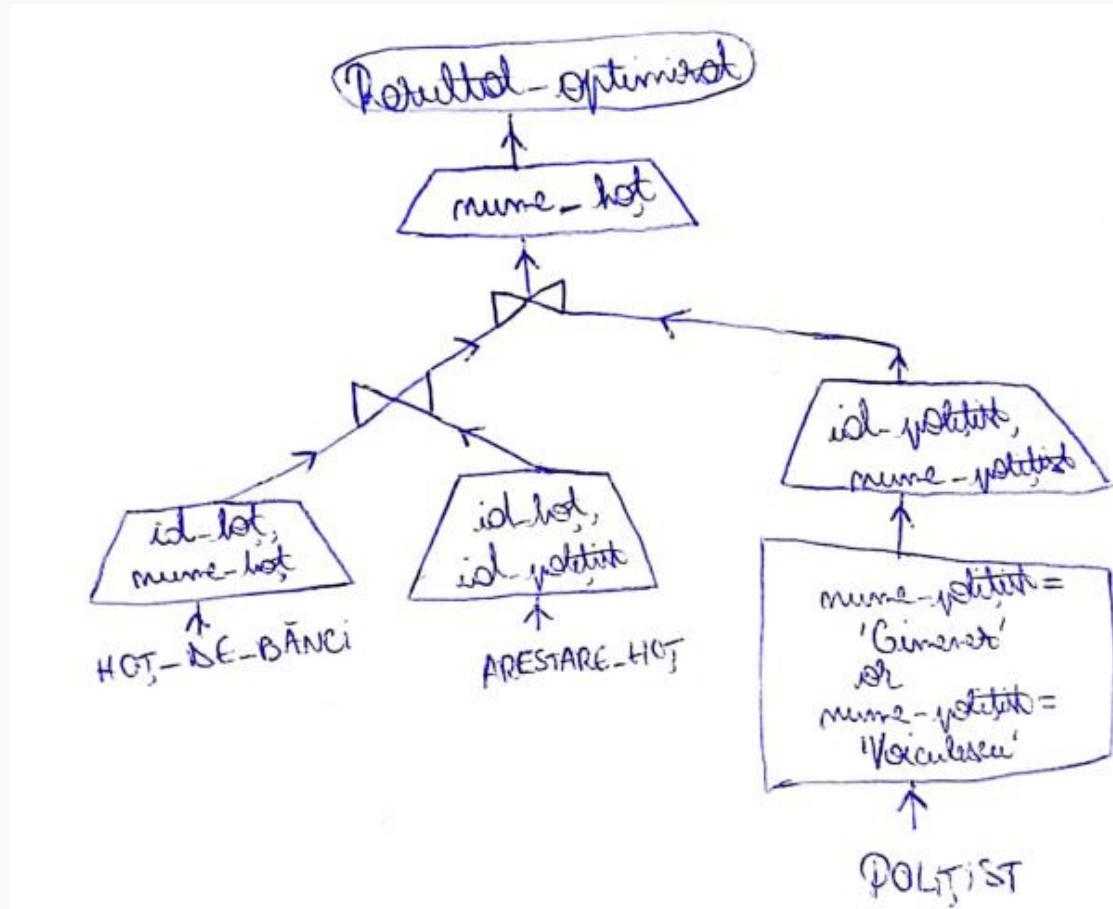
NUME_HOT
1 Armando
2 Jerome
3 Osborne
4 Spitznagel

DUPĂ OPTIMIZARE

Expresie algebraică:

Rезультат-optimizat = $\exists \text{nume_hot} (\text{JOIN}(\text{JOIN}(\exists \text{id_hot}, \text{nume_hot}(\text{HOT-DE-BĂNCI}), \exists \text{id_hot}, \text{id_politic}(\text{ARESTARE-HOT})), \exists \text{id_politic}, \text{nume_politic}(\forall \text{nume_politic} = 'Gîrbovici' \text{ or } (\text{POLITIST})))$
 $\text{nume_politic} = 'Voiculescu'$)

Arbore algebraic:



Cerere în SQL:

```
with Join11 as
    (select id_hot, nume_hot
     from hot_de_banci),
     Join12 as
    (select id_hot, id_politist
     from arestare_hot),
     SelectProject as
    (select id_politist, nume_politist
     from politist
     where nume_politist = 'Gimenez' or
           nume_politist = 'Voiculescu')
select distinct nume_hot
from Join11
join Join12 using (id_hot)
join SelectProject using(id_politist);
```

Query Result x

SQL | All Rows Fetched: 4 in 0.005 seconds

NUME_HOT
1 Spitznagel
2 Jerome
3 Armando
4 Osborne

EXPLICATIE: Pentru a fi optimizată, cererea a avut nevoie de aplicarea celor 4 reguli. Astfel, produsul cartezian a fost transformat într-un join, selecțiile au fost executate cât mai devreme posibil, iar proiecțiile au fost executate la început, pentru a fi îndepărtațe atributele nefolositoare.

17

- A. Realizarea normalizării BCNF, FN4, FN5.
- B. Aplicarea denormalizării, justificând necesitatea acesteia.

A

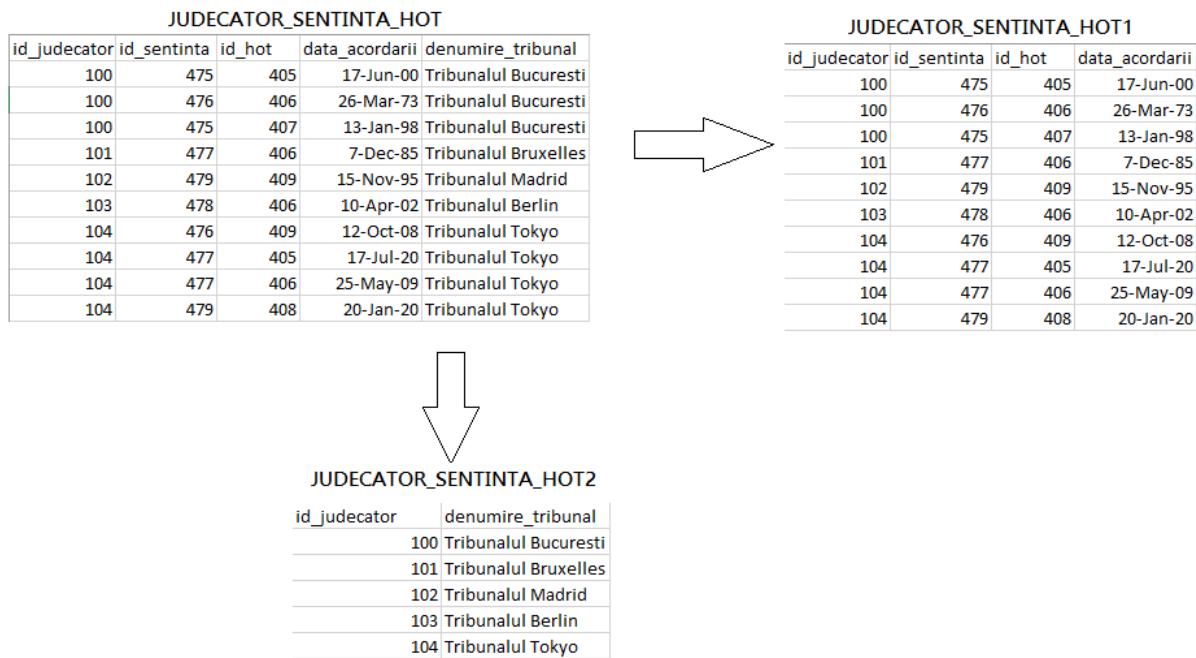
Forma normală Boyce-Codd (BCNF)

O relație R este în forma normală Boyce-Codd dacă și numai dacă pentru orice dependență funcțională totală $X \rightarrow A$, X este o cheie (candidat) a lui R (adică dacă fiecare determinant este o cheie candidat).

Modelul prezentat anterior este în FN1, FN2, FN3, BCNF.

Exemplu non-BCNF și transformarea în BCNF:

Tabelul JUDECATOR_SENTINTA_HOT:



EXPLICATIE: Tabelul JUDECATOR_SENTINTA_HOT este în FN3 deoarece toate atributele depind de o cheie și numai de o cheie. Cu toate acestea, există o redundanță, deoarece perechi de forma (id_judecator, denumire_tribunal) apar de mai multe ori în tabel. Soluția este să eliminăm redundanțele, descompunând astfel: creăm un nou tabel asemănător tabelului inițial, dar fără coloana "denumire_tribunal", apoi construim un tabel cu doar două coloane (id_judecator și denumire_tribunal), constituind împreună cheia primară a noului tabel.

Forma normală 4

O relație R este în a patra formă normală dacă și numai dacă relația este în forma normală Boyce-Codd și nu conține relații $m:n$ independente.

Modelul prezentat anterior este în FN1, FN2, FN3, BCNF, FN4.

Exemplu non-FN4 și transformarea în FN4:

Tabelul MARTOR_JAF:

The diagram illustrates the decomposition of the original table into two new tables. An arrow points from the original table to the first decomposed table, MARTOR_JAF_1. Another arrow points from MARTOR_JAF_1 to the second decomposed table, MARTOR_JAF_2.

MARTOR_JAF		
id_jaf	id_martor	id_hot
800	306	406
801	306	406
802	307	407
802	307	405
802	309	407
802	309	405
803	305	406
804	308	409
804	305	409
805	308	406
806	308	405
806	308	408
806	309	405
806	309	408

MARTOR_JAF_1	
id_jaf	id_martor
800	306
801	306
802	307
802	309
803	305
804	308
804	305
805	308
806	308
806	309

MARTOR_JAF_2	
id_jaf	id_hot
800	406
801	406
802	409
802	407
802	405
803	406
804	409
805	406
806	405
806	408

EXPLICATIE: Exemplul anterior nu încalcă nicio formă normală anterioară. Dar, deoarece hoții sunt independenți de martori, există redundanțe în tabel: de exemplu, tuplul (802, 307) apare de 2 ori în tabel. Solutia este decompunerea în două tabele, separând dependențele (id_jaf, id_martor) și (id_jaf, id_hot).

Forma normală 5

O relație R este în forma normală 5 dacă și numai dacă:

- Este în FN4
- Nu conține dependențe ciclice

Modelul prezentat este în FN1, FN2, FN3, BCNF, FN4, FN5.

Exemplu non-FN5 și transformarea în FN5:

Tabelul ARESTARE_HOT:

ARESTARE_HOT		
id_hot	id_politist	metoda_arestare
405	605	incatusare
405	609	incatusare
406	606	punere la pamant
406	607	incatusare
406	605	punere la pamant
406	609	punere la pamant
407	603	incatusare
408	609	incatusare
409	608	incatusare
409	609	punere la pamant



ARESTARE_HOT_1	
id_hot	metoda_arestare
405	incatusare
406	punere la pamant
406	incatusare
407	incatusare
408	incatusare
409	incatusare
409	punere la pamant

ARESTARE_HOT_2	
id_hot	id_politist
405	605
405	609
406	606
406	607
406	605
406	609
407	603
408	609
409	608
409	609



ARESTARE_HOT_3	
id_politist	metoda_arestare
603	incatusare
605	incatusare
605	punere la pamant
606	punere la pamant
607	incatusare
608	incatusare
609	incatusare
609	punere la pamant

EXPLICATIE: Relația este în FN4, dar există redundanță în date. Soluția este descompunerea tabelului inițial în 3 tabele, precum mai sus, astfel existând o join dependență.

Denormalizare

Înainte de normalizare

POLIȚIST(id_polițist#, nume_polițist, prenume_polițist, vârstă_polițist, data_angajării, id_secție)

SECȚIE_POLIȚIE(id_secție#, denumire_secție)

În cazul în care se dorește a fi afișată secția de poliție din care face parte un polițist, este nevoie de un join între cele două tabele.

După denormalizare

POLIȚIST(id_polițist#, nume_polițist, prenume_polițist, vârstă_polițist, data_angajării, denumire_secție)

Prin denormalizare, tabelul SECȚIE_POLIȚIE nu mai este necesar să fie reținut deoarece avea ca atrbute doar cheia primară (id-ul) și denumirea sa. Având în vedere că denormalizarea face posibila mutarea denumirii secției de poliție în tabelul POLIȚIST, join-ul nu mai este necesar. Astfel, prin denormalizare, am simplificat căutarea datelor.