Implementare bazei de date

# Crearea bazei de date si a utilizatorilor

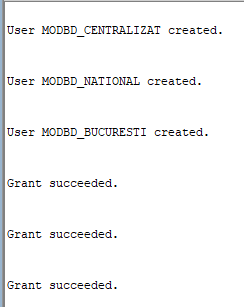
Pentru implementarea proiectului am avut nevoie de o baza de date OLTP centralizata pe care sa o distribuim in doua baze locale. Astfel am folosit doua baze de date fizice, care au fost definite pe o singura masina. Cele doua baze de date sunt MODBD1 si MODBD2.

## Utilizatori

In baza de date MODBD1 am definit 3 utilizatori:

* **modbd\_centralizat**. In schema acestui utilizator este definite baza de date centralizata.
* **modbd\_bucuresti**. In aceasta schema se afla una dintre cele doua baze de date locale.
* **modbd\_national**. In schema utlizatorului national am definit view-uri pentru a putea interactiona cu ambele baze de date locale de la nivel “global”.

Au fost creati utilizatorii si atribuite drepturi de conectare.

create user modbd\_centralizat identified by parolamodbd;

create user modbd\_national identified by parolamodbd;

create user modbd\_bucuresti identified by parolamodbd;

grant create session to modbd\_centralizat;

grant resource to modbd\_centralizat;

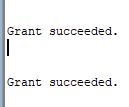
grant create session to modbd\_national;

grant resource to modbd\_national;

grant create session to modbd\_bucuresti;

grant resource to modbd\_bucuresti;

Au fost atribuite drepturi de creeare de legaturi intre baze de date utlizatorilor **modbd\_national** si **modbd\_bucuresti**:

grant CREATE PUBLIC DATABASE LINK to modbd\_national;

grant DROP PUBLIC DATABASE LINK to modbd\_national;

grant CREATE PUBLIC DATABASE LINK to modbd\_bucuresti;

grant DROP PUBLIC DATABASE LINK to modbd\_bucuresti;

Deoarece schemele utlizatorilor modbd\_national si modbd\_bucuresti sunt pe aceeasi baza de date fizica, desi BUCURESTI este considerata o baza de date locala utilizatorul modbd\_national a primit depturi asupra tabelelor lui modbd\_bucuresti.

grant select, insert, update, delete on modbd\_bucuresti.PLATA\_CHIRIE\_BUCURESTI to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.CONTRACT\_BUCURESTI to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.APARTAMENT\_BUCURESTI to modbd\_national;

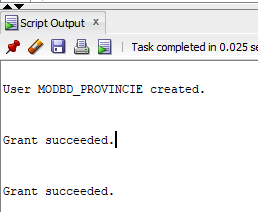
grant select, insert, update, delete on modbd\_bucuresti.AGENT\_IMOBILIAR\_BUCURESTI to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.CHIRIAS to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.JUDET\_BUCURESTI to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.LOCALITATE\_BUCURESTI to modbd\_national;

grant select, insert, update, delete on modbd\_bucuresti.ADRESA\_BUCURESTI to modbd\_national;

In baza de date MODBD2 am definit utilizatorul **modbd\_provincie**, care gazduieste cea de-a doua baza de date locala.

A fost definit utiliatorul si atribuite drepturi de conectare:

create user modbd\_provincie identified by parolamodbd;

grant create session to modbd\_provincie;

grant resource to modbd\_provincie;

De asemenea, utilizatorul modbd\_provincie a primit dreptul de a crea legaturi intre doua baze de date:



grant CREATE PUBLIC DATABASE LINK to modbd\_provincie;

## Legaturi intre bazele de date

Pentru a putea comunica intre ele, pe cele doua baze de date au fost definite 4 legaturi:

* Provincie – Bucuresti

CREATE PUBLIC DATABASE LINK BD\_BUCURESTI

CONNECT TO MODBD\_BUCURESTI

IDENTIFIED BY parolamodbd

USING '(DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))

(CONNECT\_DATA=(SERVICE\_NAME=modbd1))

)';



* Provincie – National

CREATE PUBLIC DATABASE LINK BD\_CENTRALIZAT

CONNECT TO MODBD\_CENTRALIZAT

IDENTIFIED BY parolamodbd

USING '(DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))

(CONNECT\_DATA=(SERVICE\_NAME=modbd1))

)';



* National – Provincie

CREATE PUBLIC DATABASE LINK BD\_PROVINCIE

CONNECT TO MODBD\_PROVINCIE

IDENTIFIED BY parolamodbd

USING '(DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))

(CONNECT\_DATA=(SERVICE\_NAME=modbd2))

)';



* Bucuresti – Provincie

CREATE PUBLIC DATABASE LINK BD\_PROVINCIE

CONNECT TO MODBD\_PROVINCIE

IDENTIFIED BY parolamodbd

USING '(DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))

(CONNECT\_DATA=(SERVICE\_NAME=modbd2))

)';



# Crearea relatiilor si a fragmentelor

## Crearea bazei de date centralizate

*CREATE SEQUENCE OLTP\_ADRESA\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_LOCALITATE\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_JUDET\_SEQ START WITH 1 INCREMENT BY 1;*

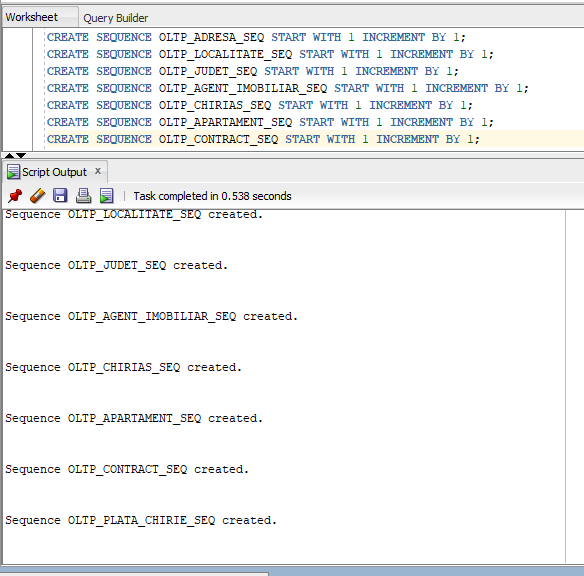
*CREATE SEQUENCE OLTP\_AGENT\_IMOBILIAR\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_CHIRIAS\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_APARTAMENT\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_CONTRACT\_SEQ START WITH 1 INCREMENT BY 1;*

*CREATE SEQUENCE OLTP\_PLATA\_CHIRIE\_SEQ START WITH 1 INCREMENT BY 1;*



**Definirea tabelelor si a constrangerilor**

create table OLTP\_JUDET

(

id\_judet NUMBER(4) DEFAULT OLTP\_JUDET\_SEQ.NEXTVAL not null,

nume VARCHAR2(25)

)

;

alter table OLTP\_JUDET

add constraint OLTP\_JUD\_ID\_PK primary key (id\_judet);

create index OLTP\_JUD\_NUME\_IX on OLTP\_JUDET (nume);

alter table OLTP\_JUDET

add constraint OLTP\_JUD\_NUME\_NN

check ("NUME" IS NOT NULL);

create table OLTP\_LOCALITATE

(

id\_localitate NUMBER(4) DEFAULT OLTP\_LOCALITATE\_SEQ.NEXTVAL not null,

nume VARCHAR2(25),

id\_judet NUMBER(4) not null

)

;

alter table OLTP\_LOCALITATE

add constraint OLTP\_LOC\_ID\_PK primary key (id\_localitate);

alter table OLTP\_LOCALITATE

add constraint OLTP\_LOC\_OLTP\_JUD\_FK foreign key (id\_judet)

references OLTP\_JUDET (id\_judet);

create index OLTP\_LOC\_NUME\_IX on OLTP\_LOCALITATE (nume);

alter table OLTP\_LOCALITATE

add constraint OLTP\_LOC\_NUME\_NN

check ("NUME" IS NOT NULL);

create table OLTP\_ADRESA

(

id\_adresa NUMBER(4) DEFAULT OLTP\_ADRESA\_SEQ.NEXTVAL not null,

strada VARCHAR2(40),

numar NUMBER(4),

bloc VARCHAR2(10),

scara VARCHAR2(5),

numar\_apartament NUMBER(4),

id\_localitate NUMBER(4) not null

)

;

alter table OLTP\_ADRESA

add constraint OLTP\_ADR\_ID\_PK primary key (id\_adresa);

alter table OLTP\_ADRESA

add constraint OLTP\_ADR\_OLTP\_LOC\_FK foreign key (id\_localitate)

references OLTP\_LOCALITATE (id\_localitate);

create index OLTP\_ADR\_STRADA\_IX on OLTP\_ADRESA (strada);

alter table OLTP\_ADRESA

add constraint OLTP\_ADR\_STRADA\_NN

check ("STRADA" IS NOT NULL);

create table OLTP\_AGENT\_IMOBILIAR

(

id\_agent NUMBER(6) DEFAULT OLTP\_AGENT\_IMOBILIAR\_SEQ.NEXTVAL not null,

prenume VARCHAR2(20),

nume VARCHAR2(25),

email VARCHAR2(25),

telefon VARCHAR2(20),

data\_angajare DATE,

salariu NUMBER(8,2),

comision NUMBER(2,2)

)

;

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_ID\_OLTP\_AGENT\_IMOBILIAR\_PK primary key (ID\_AGENT);

create index OLTP\_AGENT\_IMOBILIAR\_NUME\_IX on OLTP\_AGENT\_IMOBILIAR (NUME, PRENUME);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_EMAIL\_UK unique (EMAIL);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_TELEFON\_UK unique (TELEFON);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_DATA\_OLTP\_AGENT\_IMOBILIAR\_NN

check ("DATA\_ANGAJARE" IS NOT NULL);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_EMAIL\_NN

check ("EMAIL" IS NOT NULL);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_TELEFON\_NN

check ("TELEFON" IS NOT NULL);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_NUME\_NN

check ("NUME" IS NOT NULL);

alter table OLTP\_AGENT\_IMOBILIAR

add constraint OLTP\_AGENT\_IMOBILIAR\_SALARIU\_MIN

check (SALARIU > 0);

create table OLTP\_CHIRIAS

(

id\_chirias NUMBER(6) DEFAULT OLTP\_CHIRIAS\_SEQ.NEXTVAL not null,

prenume VARCHAR2(20),

nume VARCHAR2(20),

telefon VARCHAR2(20),

email VARCHAR2(30),

sex VARCHAR2(1),

data\_nastere DATE,

starea\_civila VARCHAR2(20)

)

;

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHIRIAS\_ID\_OLTP\_CHIRIAS\_PK primary key (ID\_CHIRIAS);

create index OLTP\_CHIRIAS\_NUME\_IX on OLTP\_CHIRIAS (NUME, PRENUME);

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHR\_NUME\_NN

check (NUME IS NOT NULL);

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHR\_PRENUME\_NN

check (PRENUME IS NOT NULL);

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHR\_EMAIL\_NN

check (EMAIL IS NOT NULL);

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHIRIAS\_EMAIL\_UK unique (EMAIL);

alter table OLTP\_CHIRIAS

add constraint OLTP\_CHIRIAS\_TELEFON\_UK unique (TELEFON);

create table OLTP\_APARTAMENT

(

id\_apartament NUMBER(4) DEFAULT OLTP\_APARTAMENT\_SEQ.NEXTVAL not null,

id\_adresa NUMBER(4) not null unique,

numar\_camere NUMBER(4),

etaj CHAR(1) not null,

suprafata NUMBER(4),

centrala\_proprie CHAR(1) not null,

pret\_inchiriere NUMBER(8, 2),

tip\_confort NUMBER(4) -- confort 1, 2, 3

);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APARTAMENT\_ID\_OLTP\_APARTAMENT\_PK primary key (ID\_APARTAMENT);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APT\_OLTP\_ADR\_FK foreign key (ID\_ADRESA)

references OLTP\_ADRESA (ID\_ADRESA);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APARTAMENT\_NR\_CAMERE\_MIN

check (numar\_camere > 0);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APARTAMENT\_SUPRAFATA\_MIN

check (suprafata > 0);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APARTAMENT\_PRET\_MIN

check (pret\_inchiriere > 0);

alter table OLTP\_APARTAMENT

add constraint OLTP\_APARTAMENT\_CONFORT\_IN

check (tip\_confort in (1, 2, 3));

create table OLTP\_CONTRACT

(

id\_contract NUMBER(4) DEFAULT OLTP\_CONTRACT\_SEQ.NEXTVAL not null,

id\_chirias NUMBER(4) not null,

id\_apartament NUMBER(4) not null,

id\_agent NUMBER(4) not null,

data\_inceput DATE not null,

data\_final DATE not null,

ziua\_scadenta NUMBER(4) not null,

pret\_inchiriere NUMBER(8, 2),

valoare\_estimata NUMBER(8, 2),

incasari NUMBER(8, 2)

);

alter table OLTP\_CONTRACT

add constraint OLTP\_CONTRACT\_AP\_PK primary key (ID\_CONTRACT);

alter table OLTP\_CONTRACT

add constraint OLTP\_CON\_AP\_ID\_FK foreign key (ID\_APARTAMENT)

references OLTP\_APARTAMENT (ID\_APARTAMENT);

alter table OLTP\_CONTRACT

add constraint OLTP\_CON\_CH\_ID\_FK foreign key (ID\_CHIRIAS)

references OLTP\_CHIRIAS (ID\_CHIRIAS);

alter table OLTP\_CONTRACT

add constraint OLTP\_CON\_AG\_ID\_FK foreign key (ID\_AGENT)

references OLTP\_AGENT\_IMOBILIAR (ID\_AGENT);

alter table OLTP\_CONTRACT

add constraint ZIUA\_SCANDETA\_CHECK

check (ziua\_scadenta >= 1 and ziua\_scadenta <= 31);

create table OLTP\_PLATA\_CHIRIE

(

id\_plata NUMBER(4) DEFAULT OLTP\_PLATA\_CHIRIE\_SEQ.NEXTVAL not null,

id\_contract NUMBER(4) not null,

luna NUMBER(4),

an NUMBER(4),

suma NUMBER(8, 2) default 0,

data\_efectuarii DATE not null,

nr\_zile\_intarziere NUMBER(2)

);

alter table OLTP\_PLATA\_CHIRIE

add constraint OLTP\_PLATA\_CHIRIE\_PK primary key (ID\_PLATA);

alter table OLTP\_PLATA\_CHIRIE

add constraint OLTP\_PLATA\_CHIRIE\_CONTRACT\_FK foreign key (ID\_CONTRACT)

references OLTP\_CONTRACT (ID\_CONTRACT);

alter table OLTP\_PLATA\_CHIRIE

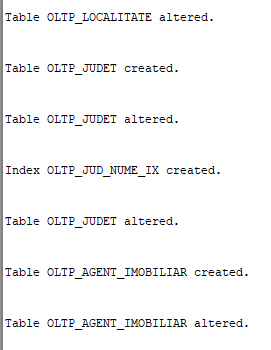
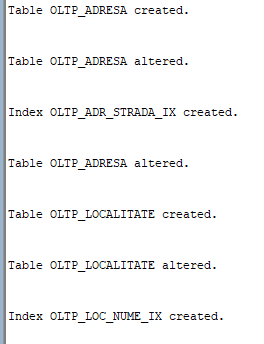
add constraint OLTP\_PLATA\_CHIRIE\_AN\_MINIM

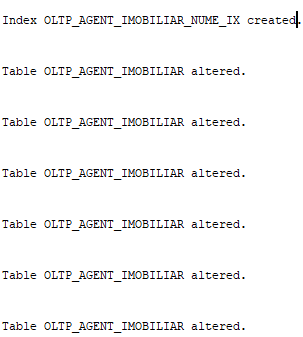
check (an > 1970);

alter table OLTP\_PLATA\_CHIRIE

add constraint OLTP\_PLATA\_CHIRIE\_VALORI\_LUNA

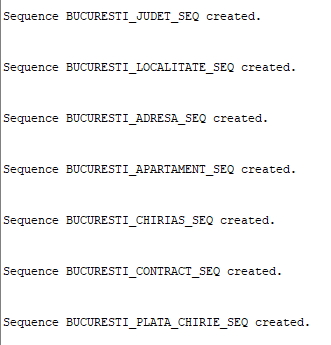
check (luna >= 1 and luna <= 12);





## Crearea bazelor de date locale

### Crearea bazei de date Bucuresti, crearea fragmentelor orizontale si stabilirea constrangerilor

 CREATE SEQUENCE BUCURESTI\_JUDET\_SEQ START WITH 43 INCREMENT BY 2;

CREATE SEQUENCE BUCURESTI\_LOCALITATE\_SEQ START WITH 149 INCREMENT BY 2;

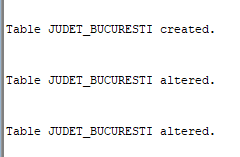
CREATE SEQUENCE BUCURESTI\_ADRESA\_SEQ START WITH 169 INCREMENT BY 2;

CREATE SEQUENCE BUCURESTI\_APARTAMENT\_SEQ START WITH 153 INCREMENT BY 2;

CREATE SEQUENCE BUCURESTI\_CHIRIAS\_SEQ START WITH 99 INCREMENT BY 2;

CREATE SEQUENCE BUCURESTI\_CONTRACT\_SEQ START WITH 55 INCREMENT BY 2;

CREATE SEQUENCE BUCURESTI\_PLATA\_CHIRIE\_SEQ START WITH 41 INCREMENT BY 2;

--JUDET\_BUCURESTI

create table JUDET\_BUCURESTI as

select \* from modbd\_centralizat.oltp\_judet

where nume = 'Bucuresti';

alter table JUDET\_BUCURESTI

add constraint JUD\_BUC\_ID\_PK primary key (id\_judet);

alter table JUDET\_BUCURESTI

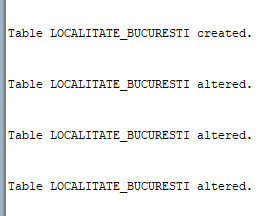
add constraint JUD\_BUCURESTI\_NUME\_NN

check ("NUME" IS NOT NULL);

--LOCALITATE\_BUCURESTI

create table LOCALITATE\_BUCURESTI as

select loc.\*

from modbd\_centralizat.oltp\_localitate loc

join modbd\_centralizat.oltp\_judet jud on loc.id\_judet = jud.id\_judet

where jud.nume = 'Bucuresti';

alter table LOCALITATE\_BUCURESTI

add constraint LOC\_BUC\_ID\_PK primary key (id\_localitate);

alter table LOCALITATE\_BUCURESTI

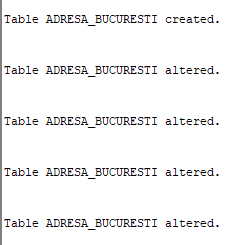
add constraint LOC\_BUC\_JUD\_FK foreign key (id\_judet)

references JUDET\_BUCURESTI (id\_judet);

alter table LOCALITATE\_BUCURESTI

add constraint LOC\_BUC\_NUME\_NN check ("NUME" IS NOT NULL);

--ADRESA\_BUCURESTI

create table ADRESA\_BUCURESTI as

select adr.\*

from modbd\_centralizat.oltp\_adresa adr

join modbd\_centralizat.oltp\_localitate loc on adr.id\_localitate = loc.id\_localitate

join modbd\_centralizat.oltp\_judet jud on loc.id\_judet = jud.id\_judet

where jud.nume = 'Bucuresti';

alter table ADRESA\_BUCURESTI

add constraint ADR\_BUC\_ID\_PK primary key (id\_adresa);

alter table ADRESA\_BUCURESTI

add constraint ADR\_BUC\_LOC\_FK foreign key (id\_localitate)

references LOCALITATE\_BUCURESTI (id\_localitate);

alter table ADRESA\_BUCURESTI

add constraint ADR\_BUC\_STRADA\_NN

check ("STRADA" IS NOT NULL);

alter table adresa\_BUCURESTI

add constraint ADR\_BUC\_UNIQUE UNIQUE (STRADA, NUMAR, BLOC, SCARA, NUMAR\_APARTAMENT);

--APARTAMENT\_BUCURESTI

create table APARTAMENT\_BUCURESTI as

select ap.\*

from modbd\_centralizat.oltp\_apartament ap

join modbd\_centralizat.oltp\_adresa adr on ap.id\_adresa = adr.id\_adresa

join modbd\_centralizat.oltp\_localitate loc on adr.id\_localitate = loc.id\_localitate

join modbd\_centralizat.oltp\_judet jud on loc.id\_judet = jud.id\_judet

where jud.nume = 'Bucuresti';

alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_ID\_BUC\_APARTAMENT\_PK primary key (ID\_APARTAMENT);

alter table APARTAMENT\_BUCURESTI

add constraint APT\_BUC\_\_ADR\_FK foreign key (ID\_ADRESA)

references ADRESA\_BUCURESTI (ID\_ADRESA);

alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_BUC\_NR\_CAMERE\_MIN

check (numar\_camere > 0);

alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_BUC\_SUPRAFATA\_MIN

check (suprafata > 0);

alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_BUC\_PRET\_MIN

check (pret\_inchiriere > 0);

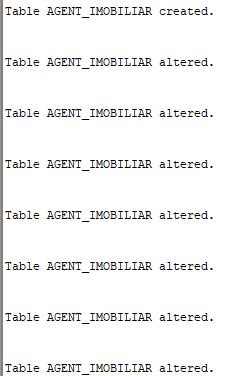
alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_BUC\_CONFORT\_IN

check (tip\_confort in (1, 2, 3));

alter table APARTAMENT\_BUCURESTI

add constraint APARTAMENT\_BUC\_ADR\_UK UNIQUE (ID\_ADRESA);

--agent imobiliar

create table agent\_imobiliar as

select id\_agent, prenume, nume, email, telefon, data\_angajare

from modbd\_centralizat.oltp\_agent\_imobiliar;

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_ID\_AGENT\_IMOBILIAR\_PK primary key (ID\_AGENT);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_EMAIL\_UK unique (EMAIL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_TELEFON\_UK unique (TELEFON);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_DATA\_AGENT\_IMOBILIAR\_NN

check ("DATA\_ANGAJARE" IS NOT NULL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_EMAIL\_NN

check ("EMAIL" IS NOT NULL);

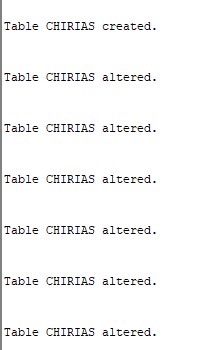
alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_TELEFON\_NN

check ("TELEFON" IS NOT NULL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_NUME\_NN check ("NUME" IS NOT NULL);

-- CHIRIAS

create table chirias as

select \* from modbd\_centralizat.oltp\_chirias;

alter table CHIRIAS

add constraint CHIRIAS\_ID\_CHIRIAS\_PK primary key (ID\_CHIRIAS);

alter table CHIRIAS

add constraint CHR\_NUME\_NN

check (NUME IS NOT NULL); alter table CHIRIAS

add constraint CHR\_PRENUME\_NN

check (PRENUME IS NOT NULL);

alter table CHIRIAS

add constraint CHR\_EMAIL\_NN

check (EMAIL IS NOT NULL);

alter table CHIRIAS

add constraint CHIRIAS\_EMAIL\_UK unique (EMAIL);

alter table CHIRIAS

add constraint CHIRIAS\_TELEFON\_UK unique (TELEFON);

--CONTRACT\_BUCURESTI

create table CONTRACT\_BUCURESTI AS

SELECT ID\_CONTRACT, ID\_CHIRIAS, con.ID\_APARTAMENT, ID\_AGENT, DATA\_INCEPUT, DATA\_FINAL, ZIUA\_SCADENTA, con.PRET\_INCHIRIERE, VALOARE\_ESTIMATA, INCASARI

FROM modbd\_centralizat.oltp\_contract con

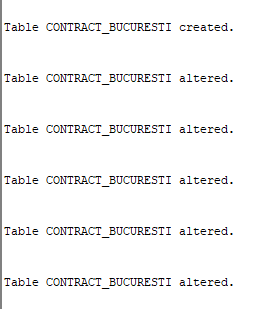
join modbd\_centralizat.oltp\_apartament ap on con.id\_apartament = ap.id\_apartament

join modbd\_centralizat.oltp\_adresa adr on ap.id\_adresa = adr.id\_adresa

join modbd\_centralizat.oltp\_localitate loc on adr.id\_localitate = loc.id\_localitate

where loc.nume = 'Bucuresti';

alter table CONTRACT\_BUCURESTI

 add constraint CONTRACT\_BUC\_AP\_PK primary key (ID\_CONTRACT);

alter table CONTRACT\_BUCURESTI

add constraint CON\_BUC\_AP\_ID\_FK foreign key (ID\_APARTAMENT)

references APARTAMENT\_BUCURESTI (ID\_APARTAMENT);

alter table CONTRACT\_BUCURESTI

add constraint CONTRACT\_BUC\_\_CH\_ID\_FK foreign key (ID\_CHIRIAS)

references CHIRIAS (ID\_CHIRIAS);

alter table CONTRACT\_BUCURESTI

add constraint CONTRACT\_BUC\_AG\_ID\_FK foreign key (ID\_AGENT)

references AGENT\_IMOBILIAR (ID\_AGENT);

alter table CONTRACT\_BUCURESTI

add constraint ZIUA\_SCANDETA\_CHECK

check (ziua\_scadenta >= 1 and ziua\_scadenta <= 31);

--PLATA\_CHIRIE\_BUCURESTI

create table PLATA\_CHIRIE\_BUCURESTI as

SELECT ID\_PLATA, pc.ID\_CONTRACT, LUNA, AN, SUMA, DATA\_EFECTUARII, NR\_ZILE\_INTARZIERE

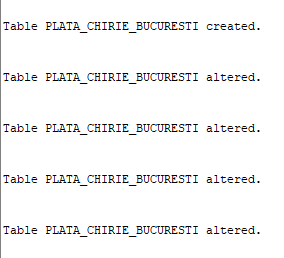
FROM modbd\_centralizat.OLTP\_PLATA\_CHIRIE pc

join modbd\_centralizat.oltp\_contract con on pc.id\_contract = con.id\_contract

join modbd\_centralizat.oltp\_apartament ap on con.id\_apartament = ap.id\_apartament

join modbd\_centralizat.oltp\_adresa adr on ap.id\_adresa = adr.id\_adresa

join modbd\_centralizat.oltp\_localitate loc on adr.id\_localitate = loc.id\_localitate

where loc.nume = 'Bucuresti';

alter table PLATA\_CHIRIE\_BUCURESTI

add constraint PLATA\_CHIRIE\_BUC\_CHIRIE\_PK primary key (ID\_PLATA);

alter table PLATA\_CHIRIE\_BUCURESTI

add constraint PLATA\_CHIRIE\_BUC\_CHIRIE\_CONTRACT\_FK foreign key (ID\_CONTRACT)

references CONTRACT\_BUCURESTI (ID\_CONTRACT);

alter table PLATA\_CHIRIE\_BUCURESTI

add constraint PLATA\_CHIRIE\_BUC\_CHIRIE\_AN\_MINIM

check (an > 1970);

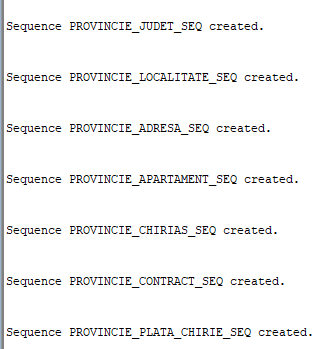
alter table PLATA\_CHIRIE\_BUCURESTI

add constraint PLATA\_CHIRIE\_BUC\_CHIRIE\_VALORI\_LUNA

check (luna >= 1 and luna <= 12);

### Crearea bazei de date Provincie, crearea fragmentelor orizontale si stabilirea constrangerilor

CREATE SEQUENCE PROVINCIE\_JUDET\_SEQ START WITH 44 INCREMENT BY 2;

 CREATE SEQUENCE PROVINCIE\_LOCALITATE\_SEQ START WITH 148 INCREMENT BY 2;

CREATE SEQUENCE PROVINCIE\_ADRESA\_SEQ START WITH 168 INCREMENT BY 2;

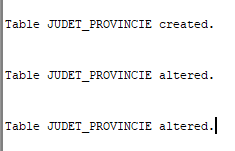
CREATE SEQUENCE PROVINCIE\_APARTAMENT\_SEQ START WITH 152 INCREMENT BY 2;

CREATE SEQUENCE PROVINCIE\_CHIRIAS\_SEQ START WITH 98 INCREMENT BY 2;

CREATE SEQUENCE PROVINCIE\_CONTRACT\_SEQ START WITH 56 INCREMENT BY 2;

CREATE SEQUENCE PROVINCIE\_PLATA\_CHIRIE\_SEQ START WITH 42 INCREMENT BY 2;

--JUDET\_PROVINCIE

create table JUDET\_PROVINCIE as

select \* from oltp\_judet@BD\_CENTRALIZAT

where nume <> 'Bucuresti';

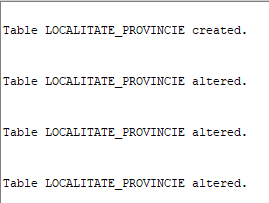
alter table JUDET\_PROVINCIE

add constraint JUD\_PROV\_ID\_PK primary key (id\_judet);

alter table JUDET\_PROVINCIE

add constraint JUD\_PROVINCIE\_NUME\_NN

check ("NUME" IS NOT NULL);

--LOCALITATE\_PROVINCIE

create table LOCALITATE\_PROVINCIE as

select loc.\*

from oltp\_localitate@BD\_CENTRALIZAT loc

join oltp\_judet@BD\_CENTRALIZAT jud on loc.id\_judet = jud.id\_judet

where jud.nume <> 'Bucuresti';

alter table LOCALITATE\_PROVINCIE

add constraint LOC\_PROV\_ID\_PK primary key (id\_localitate);

alter table LOCALITATE\_PROVINCIE

add constraint LOC\_PROV\_JUD\_FK foreign key (id\_judet)

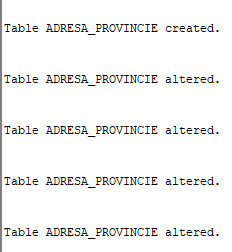
references JUDET\_PROVINCIE (id\_judet);

alter table LOCALITATE\_PROVINCIE

add constraint LOC\_PROV\_NUME\_NN

check ("NUME" IS NOT NULL);

--ADRESA\_PROVINCIE

create table ADRESA\_PROVINCIE as

select adr.\*

from oltp\_adresa@BD\_CENTRALIZAT adr

join oltp\_localitate@BD\_CENTRALIZAT loc on adr.id\_localitate = loc.id\_localitate

join oltp\_judet@BD\_CENTRALIZAT jud on loc.id\_judet = jud.id\_judet

where jud.nume <> 'Bucuresti';

alter table ADRESA\_PROVINCIE

add constraint ADR\_PROV\_ID\_PK primary key (id\_adresa);

alter table ADRESA\_PROVINCIE

add constraint ADR\_PROV\_LOC\_FK foreign key (id\_localitate)

references LOCALITATE\_PROVINCIE (id\_localitate);

alter table ADRESA\_PROVINCIE

add constraint ADR\_PROV\_STRADA\_NN

check ("STRADA" IS NOT NULL);

alter table adresa\_provincie

add constraint ADR\_PROV\_UNIQUE UNIQUE (STRADA, NUMAR, BLOC, SCARA, NUMAR\_APARTAMENT);

--APARTAMENT\_PROVINCIE

create table APARTAMENT\_PROVINCIE as

select ap.\*

from oltp\_apartament@BD\_CENTRALIZAT ap

join oltp\_adresa@BD\_CENTRALIZAT adr on ap.id\_adresa = adr.id\_adresa

join oltp\_localitate@BD\_CENTRALIZAT loc on adr.id\_localitate = loc.id\_localitate

join oltp\_judet@BD\_CENTRALIZAT jud on loc.id\_judet = jud.id\_judet

where jud.nume <> 'Bucuresti';

alter table APARTAMENT\_PROVINCIE

add constraint APARTAMENT\_ID\_PROV\_APARTAMENT\_PK primary key (ID\_APARTAMENT);

alter table APARTAMENT\_PROVINCIE

add constraint APT\_PROV\_\_ADR\_FK foreign key (ID\_ADRESA)

references ADRESA\_PROVINCIE (ID\_ADRESA);

alter table APARTAMENT\_PROVINCIE

add constraint APARTAMENT\_PROV\_NR\_CAMERE\_MIN

check (numar\_camere > 0);

alter table APARTAMENT\_PROVINCIE

add constraint APARTAMENT\_PROV\_SUPRAFATA\_MIN

check (suprafata > 0);

alter table APARTAMENT\_PROVINCIE

add constraint APARTAMENT\_PROV\_PRET\_MIN

check (pret\_inchiriere > 0);

alter table APARTAMENT\_PROVINCIE

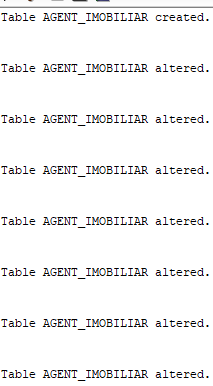
add constraint APARTAMENT\_PROV\_CONFORT\_IN

check (tip\_confort in (1, 2, 3));

alter table APARTAMENT\_PROVINCIE

add constraint APARTAMENT\_PROV\_ADR\_UK UNIQUE (ID\_ADRESA);

--agent\_imobiliar

create table agent\_imobiliar as

select id\_agent, prenume, nume, email, telefon, data\_angajare

from oltp\_agent\_imobiliar@BD\_CENTRALIZAT;

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_ID\_AGENT\_IMOBILIAR\_PK primary key (ID\_AGENT);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_EMAIL\_UK unique (EMAIL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_TELEFON\_UK unique (TELEFON);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_DATA\_AGENT\_IMOBILIAR\_NN

check ("DATA\_ANGAJARE" IS NOT NULL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_EMAIL\_NN

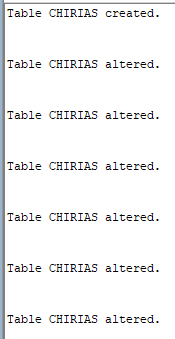
check ("EMAIL" IS NOT NULL);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_TELEFON\_NN check ("TELEFON" IS NOT NULL);

alter table AGENT\_IMOBILIAR add constraint AGENT\_IMOBILIAR\_NUME\_NN check ("NUME" IS NOT NULL);

-- CHIRIAS

create table chirias as

select \* from oltp\_chirias@BD\_CENTRALIZAT;

alter table CHIRIAS

add constraint CHIRIAS\_ID\_CHIRIAS\_PK primary key (ID\_CHIRIAS);

alter table CHIRIAS

add constraint CHR\_NUME\_NN

check (NUME IS NOT NULL);

alter table CHIRIAS

add constraint CHR\_PRENUME\_NN

check (PRENUME IS NOT NULL);

alter table CHIRIAS

add constraint CHR\_EMAIL\_NN

check (EMAIL IS NOT NULL);

alter table CHIRIAS

add constraint CHIRIAS\_EMAIL\_UK unique (EMAIL);

alter table CHIRIAS

add constraint CHIRIAS\_TELEFON\_UK unique (TELEFON);

--CONTRACT\_PROVINCIE

create table CONTRACT\_PROVINCIE AS

SELECT ID\_CONTRACT, ID\_CHIRIAS, con.ID\_APARTAMENT, ID\_AGENT, DATA\_INCEPUT, DATA\_FINAL, ZIUA\_SCADENTA, con.PRET\_INCHIRIERE, VALOARE\_ESTIMATA, INCASARI

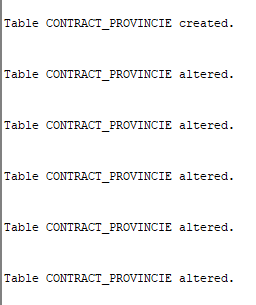
FROM oltp\_contract@BD\_CENTRALIZAT con

join oltp\_apartament@BD\_CENTRALIZAT ap on con.id\_apartament = ap.id\_apartament

join oltp\_adresa@BD\_CENTRALIZAT adr on ap.id\_adresa = adr.id\_adresa

join oltp\_localitate@BD\_CENTRALIZAT loc on adr.id\_localitate = loc.id\_localitate

where loc.nume <> 'Bucuresti';

alter table CONTRACT\_PROVINCIE

add constraint CONTRACT\_PROV\_AP\_PK primary key (ID\_CONTRACT);

alter table CONTRACT\_PROVINCIE

add constraint CON\_PROV\_AP\_ID\_FK foreign key (ID\_APARTAMENT)

references APARTAMENT\_PROVINCIE (ID\_APARTAMENT);

alter table CONTRACT\_PROVINCIE

add constraint CONTRACT\_PROV\_\_CH\_ID\_FK foreign key (ID\_CHIRIAS)

references CHIRIAS (ID\_CHIRIAS);

alter table CONTRACT\_PROVINCIE

add constraint CONTRACT\_PROV\_AG\_ID\_FK foreign key (ID\_AGENT)

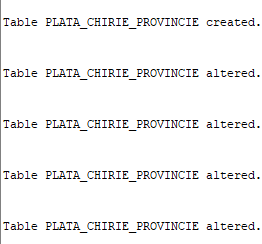
references AGENT\_IMOBILIAR (ID\_AGENT);

alter table CONTRACT\_PROVINCIE

add constraint ZIUA\_SCANDETA\_CHECK

check (ziua\_scadenta >= 1 and ziua\_scadenta <= 31);

--PLATA\_CHIRIE\_PROVINCIE

create table PLATA\_CHIRIE\_PROVINCIE as

SELECT ID\_PLATA, pc.ID\_CONTRACT, LUNA, AN, SUMA, DATA\_EFECTUARII, NR\_ZILE\_INTARZIERE

FROM OLTP\_PLATA\_CHIRIE@BD\_CENTRALIZAT pc

join oltp\_contract@BD\_CENTRALIZAT con on pc.id\_contract = con.id\_contract

join oltp\_apartament@BD\_CENTRALIZAT ap on con.id\_apartament = ap.id\_apartament

join oltp\_adresa@BD\_CENTRALIZAT adr on ap.id\_adresa = adr.id\_adresa

join oltp\_localitate@BD\_CENTRALIZAT loc on adr.id\_localitate = loc.id\_localitate

where loc.nume <> 'Bucuresti';

alter table PLATA\_CHIRIE\_PROVINCIE

add constraint PLATA\_CHIRIE\_PROV\_CHIRIE\_PK primary key (ID\_PLATA);

alter table PLATA\_CHIRIE\_PROVINCIE

add constraint PLATA\_CHIRIE\_PROV\_CHIRIE\_CONTRACT\_FK foreign key (ID\_CONTRACT)

references CONTRACT\_PROVINCIE (ID\_CONTRACT);

alter table PLATA\_CHIRIE\_PROVINCIE

add constraint PLATA\_CHIRIE\_PROV\_CHIRIE\_AN\_MINIM

check (an > 1970);

alter table PLATA\_CHIRIE\_PROVINCIE

add constraint PLATA\_CHIRIE\_PROV\_CHIRIE\_VALORI\_LUNA

check (luna >= 1 and luna <= 12);

### Fragmentarea tabelului OLTP\_AGENT\_IMOBILIAR

Dupa cum se poate observa din create-urile de mai sus pentru tabelele AGENT\_IMOBILIAR din Provincie sau Bucuresti, lipsesc coloanele SALARIU si COMISION. Aceste date sunt stocate in schema utilizatorului modbd\_national, si reprezinta un fragment vertical al tabelului OLTP\_AGENT\_IMOBILIAR.

create table AGENT\_IMOBILIAR as

select id\_agent, salariu, comision from modbd\_centralizat.oltp\_agent\_imobiliar;

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_ID\_AGENT\_IMOBILIAR\_NATIONAL\_PK primary key (ID\_AGENT);

alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_SALARIU\_MIN

check (SALARIU > 0);

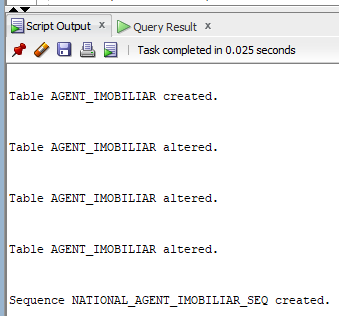
alter table AGENT\_IMOBILIAR

add constraint AGENT\_IMOBILIAR\_COMISION\_MIN

check (comision >= 0);

Pentru a genera id-uri unice la nivel de angent se va folosit doar o secventa definite la nivel global.

CREATE SEQUENCE NATIONAL\_AGENT\_IMOBILIAR\_SEQ START WITH 72 INCREMENT BY 1;



# Popularea cu date a bazelor de date

**Popularea bazei de date centralizata.**

Pentru baza de date centralizata au fost definite insert-uri pentru fiecare tabel in parte care folosesc secvente pentru generarea id-urilor unice.

insert into oltp\_AGENT\_IMOBILIAR values (oltp\_AGENT\_IMOBILIAR\_SEQ.NEXTVAL, 'Andrei', 'Popa', 'george\_1@gmail.com', '0723456789', to\_date('12.05.2001', 'dd-mm-yyyy'), 3000, 0.15);

insert into oltp\_CHIRIAS values (oltp\_CHIRIAS\_SEQ.NEXTVAL, 'Cristian', 'Olteanu', '0712345678', 'george@gmail.com', 'M', to\_date('12.05.2000', 'dd-mm-yyyy'), 'singur');

INSERT INTO oltp\_JUDET VALUES (oltp\_JUDET\_SEQ.NEXTVAL, 'Alba');

INSERT INTO oltp\_LOCALITATE VALUES (oltp\_LOCALITATE\_SEQ.NEXTVAL, 'Alba Iulia', 1);

insert into oltp\_ADRESA values (oltp\_ADRESA\_SEQ.NEXTVAL, 'Strada Primaverii', 12, 'Bl. C', 'Sc. A', 3, 2);

insert into oltp\_APARTAMENT values (oltp\_APARTAMENT\_SEQ.NEXTVAL, 1, 2, '1', 65, 'Y', 2500, 2);

INSERT into oltp\_CONTRACT values (oltp\_CONTRACT\_SEQ.NEXTVAL, 34, 23, 12, to\_date('01-01-2002', 'dd-mm-yyyy'), to\_date('31-12-2002', 'dd-mm-yyyy'), 5, null, null, 0);

INSERT INTO oltp\_plata\_chirie VALUES (oltp\_plata\_chirie\_SEQ.NEXTVAL, 1, 1, 2000, 1800, to\_date('01-01-2000', 'dd-mm-yyyy'), 1);

**Popularea bazelor de date locale.**

Popularea fragmentelor bazei de date centralizate se face la fragmentarea tabelelor din baza de date centralizata.

# Furnizarea formelor de transparenta

## Transparenta pentru fragmentele verticale

Tabelul OLTP\_AGENT\_IMOBILIAR a fost fragmentat vertical pentru a separa datele personale ale agentului de detaliile despre salarizare.

Pentru a pastra ideea ca utilizatorul interactioneaza cu o singura entitate a fost definit un view care furnizeaza date din cele doua fragmente ale tabelului.

create or replace view oltp\_AGENT\_IMOBILIAR

as

select ag\_prov.id\_agent, ag\_prov.nume, ag\_prov.prenume, ag\_prov.email, ag\_prov.telefon, ag\_prov.data\_angajare, ag\_nat\_1.salariu, ag\_nat\_1.comision

from modbd\_provincie.AGENT\_IMOBILIAR@bd\_provincie ag\_prov

join agent\_imobiliar ag\_nat\_1 on ag\_prov.id\_agent = ag\_nat\_1.id\_agent;



Garantarea sincronizarii datelor se face prin intermediul unui trigger definit in schema utilizatorului modbd\_national.

CREATE OR REPLACE TRIGGER t\_agent

INSTEAD OF INSERT OR UPDATE OR DELETE ON oltp\_agent\_imobiliar

FOR EACH ROW

DECLARE

id\_agent\_nextval NUMBER;

BEGIN

id\_agent\_nextval := modbd\_national.NATIONAL\_AGENT\_IMOBILIAR\_SEQ.NEXTVAL;

IF INSERTING THEN

BEGIN

INSERT INTO agent\_imobiliar@bd\_provincie VALUES (id\_agent\_nextval, :NEW.prenume, :NEW.nume, :NEW.email, :NEW.telefon, :NEW.data\_angajare);

INSERT INTO modbd\_bucuresti.agent\_imobiliar VALUES (id\_agent\_nextval, :NEW.prenume, :NEW.nume, :NEW.email, :NEW.telefon, :NEW.data\_angajare);

INSERT INTO modbd\_national.agent\_imobiliar VALUES (id\_agent\_nextval, :NEW.salariu, :NEW.comision);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE agent\_imobiliar@bd\_provincie SET prenume = :NEW.prenume, nume = :NEW.nume, email = :NEW.email, telefon = :NEW.telefon, data\_angajare = :NEW.data\_angajare

WHERE id\_agent = :NEW.id\_agent AND (prenume <> :NEW.prenume OR nume <> :NEW.nume OR email <> :NEW.email OR telefon <> :NEW.telefon OR data\_angajare <> :NEW.data\_angajare) ;

UPDATE modbd\_bucuresti.agent\_imobiliar SET prenume = :NEW.prenume, nume = :NEW.nume, email = :NEW.email, telefon = :NEW.telefon, data\_angajare = :NEW.data\_angajare

WHERE id\_agent = :NEW.id\_agent AND (prenume <> :NEW.prenume OR nume <> :NEW.nume OR email <> :NEW.email OR telefon <> :NEW.telefon OR data\_angajare <> :NEW.data\_angajare) ;

UPDATE modbd\_national.agent\_imobiliar SET salariu = :NEW.salariu, comision = :NEW.comision

WHERE id\_agent = :NEW.id\_agent AND (salariu <> :NEW.salariu OR comision <> :NEW.comision) ;

ELSIF DELETING THEN

DELETE FROM agent\_imobiliar@bd\_provincie WHERE id\_agent = :OLD.id\_agent;

DELETE FROM modbd\_bucuresti.agent\_imobiliar WHERE id\_agent = :OLD.id\_agent;

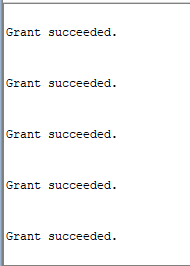
DELETE FROM modbd\_national.agent\_imobiliar WHERE id\_agent = :OLD.id\_agent;

END IF;

END;

/

Pentru a putea implementa transparenta atat pe fragmentele orizontale cat si pentru relatiile replicate au trebuit atribuite drepturi utilizatorului modbd\_national pentru secventele utilizatorului modbd\_bucuresti.

grant select on modbd\_bucuresti.BUCURESTI\_CHIRIAS\_SEQ to modbd\_national;

grant select on modbd\_bucuresti.BUCURESTI\_ADRESA\_SEQ to modbd\_national;

grant select on modbd\_bucuresti.BUCURESTI\_APARTAMENT\_SEQ to modbd\_national;

grant select on modbd\_bucuresti.BUCURESTI\_CONTRACT\_SEQ to modbd\_national;

grant select on modbd\_bucuresti.BUCURESTI\_PLATA\_CHIRIE\_SEQ to modbd\_national;

## Transparenta pentru fragmentele orizontale

Pentru relatiile  **OLTP\_JUDET** si **OLTP\_LOCALITATE** am definit doar view-uri la nivel local, fara a mai defini si triggeri instead of, doarece nu se doreste modificarea tabelelor de la nivel global.

create or replace view oltp\_judet

as

select \* from modbd\_provincie.judet\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.judet\_bucuresti;

create or replace view oltp\_localitate

as

select \* from modbd\_provincie.localitate\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.localitate\_bucuresti;

Transparenta pentru **OLTP\_ADRESA**.

create or replace view oltp\_adresa

as

select \* from modbd\_provincie.adresa\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.adresa\_bucuresti;

A fost creat un trigger instead of insert or update or delete pentru view-ul creat pentru a mentine ideea ca utilizatorul interactioneaza cu o singura entitate.



CREATE OR REPLACE TRIGGER t\_adresa

INSTEAD OF INSERT OR UPDATE OR DELETE ON oltp\_adresa

FOR EACH ROW

DECLARE

v\_in\_bucuresti NUMBER := 0;

BEGIN

select count (\*) into v\_in\_bucuresti from modbd\_bucuresti.localitate\_bucuresti loc

join modbd\_bucuresti.judet\_bucuresti jud on loc.id\_judet = jud.id\_judet

where id\_localitate = :new.id\_localitate AND jud.nume = 'Bucuresti';

if v\_in\_bucuresti = 0 then

IF INSERTING THEN

BEGIN

INSERT INTO adresa\_provincie@bd\_provincie VALUES (PROVINCIE\_ADRESA\_SEQ.NEXTVAL@bd\_provincie, :NEW.strada, :NEW.numar, :NEW.bloc, :NEW.scara, :NEW.numar\_apartament, :NEW.id\_localitate);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE adresa\_provincie@bd\_provincie SET id\_adresa = :NEW.id\_adresa, strada = :NEW.strada, numar = :NEW.numar, bloc = :NEW.bloc, scara = :NEW.scara, numar\_apartament = :NEW.numar\_apartament, id\_localitate = :NEW.id\_localitate

WHERE id\_adresa = :NEW.id\_adresa AND (strada <> :NEW.strada OR numar <> :NEW.numar OR bloc <> :NEW.bloc OR scara <> :NEW.scara OR numar\_apartament <> :NEW.numar\_apartament OR id\_localitate <> :NEW.id\_localitate ) ;

ELSIF DELETING THEN

DELETE FROM adresa\_provincie@bd\_provincie WHERE id\_adresa = :OLD.id\_adresa;

END IF;

else

IF INSERTING THEN

BEGIN

INSERT INTO modbd\_bucuresti.adresa\_bucuresti VALUES (modbd\_bucuresti.BUCURESTI\_ADRESA\_SEQ.NEXTVAL, :NEW.strada, :NEW.numar, :NEW.bloc, :NEW.scara, :NEW.numar\_apartament, :NEW.id\_localitate);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE modbd\_bucuresti.adresa\_bucuresti SET id\_adresa = :NEW.id\_adresa, strada = :NEW.strada, numar = :NEW.numar, bloc = :NEW.bloc, scara = :NEW.scara, numar\_apartament = :NEW.numar\_apartament, id\_localitate = :NEW.id\_localitate

WHERE id\_adresa = :NEW.id\_adresa AND (strada <> :NEW.strada OR numar <> :NEW.numar OR bloc <> :NEW.bloc OR scara <> :NEW.scara OR numar\_apartament <> :NEW.numar\_apartament OR id\_localitate <> :NEW.id\_localitate ) ;

ELSIF DELETING THEN

DELETE FROM modbd\_bucuresti.adresa\_bucuresti WHERE id\_adresa = :OLD.id\_adresa;

END IF;

end if;

END;

Transaparenta pentru **OLTP\_APARTAMENT.**

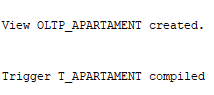
create or replace view oltp\_apartament

as

select \* from modbd\_provincie.apartament\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.apartament\_bucuresti;



CREATE OR REPLACE TRIGGER t\_apartament

INSTEAD OF INSERT OR UPDATE OR DELETE ON oltp\_apartament

FOR EACH ROW

DECLARE

v\_in\_bucuresti NUMBER := 0;

BEGIN

select count (\*) into v\_in\_bucuresti from modbd\_bucuresti.adresa\_bucuresti adr

join modbd\_bucuresti.localitate\_bucuresti loc on adr.id\_localitate = loc.id\_localitate

join modbd\_bucuresti.judet\_bucuresti jud on loc.id\_judet = jud.id\_judet

where adr.id\_adresa = :NEW.id\_adresa AND jud.nume = 'Bucuresti';

if v\_in\_bucuresti = 0 then

IF INSERTING THEN

BEGIN

INSERT INTO apartament\_provincie@bd\_provincie VALUES (PROVINCIE\_APARTAMENT\_SEQ.NEXTVAL@bd\_provincie, :NEW.id\_adresa, :NEW.numar\_camere, :NEW.etaj, :NEW.suprafata, :NEW.centrala\_proprie, :NEW.pret\_inchiriere, :NEW.tip\_confort);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE apartament\_provincie@bd\_provincie SET id\_adresa = :NEW.id\_adresa, numar\_camere = :NEW.numar\_camere, etaj = :NEW.etaj, suprafata = :NEW.suprafata, centrala\_proprie = :NEW.centrala\_proprie, pret\_inchiriere = :NEW.pret\_inchiriere, tip\_confort = :NEW.tip\_confort

WHERE id\_apartament = :NEW.id\_apartament AND (id\_adresa <> :NEW.id\_adresa OR numar\_camere <> :NEW.numar\_camere OR etaj <> :NEW.etaj OR suprafata <> :NEW.suprafata OR centrala\_proprie <> :NEW.centrala\_proprie OR pret\_inchiriere <> :NEW.pret\_inchiriere OR tip\_confort <> :NEW.tip\_confort ) ;

ELSIF DELETING THEN

DELETE FROM apartament\_provincie@bd\_provincie WHERE id\_apartament = :OLD.id\_apartament;

END IF;

else

IF INSERTING THEN

BEGIN

INSERT INTO modbd\_bucuresti.apartament\_bucuresti VALUES (modbd\_bucuresti.BUCURESTI\_APARTAMENT\_SEQ.NEXTVAL, :NEW.id\_adresa, :NEW.numar\_camere, :NEW.etaj, :NEW.suprafata, :NEW.centrala\_proprie, :NEW.pret\_inchiriere, :NEW.tip\_confort);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE modbd\_bucuresti.apartament\_bucuresti SET id\_adresa = :NEW.id\_adresa, numar\_camere = :NEW.numar\_camere, etaj = :NEW.etaj, suprafata = :NEW.suprafata, centrala\_proprie = :NEW.centrala\_proprie, pret\_inchiriere = :NEW.pret\_inchiriere, tip\_confort = :NEW.tip\_confort

WHERE id\_apartament = :NEW.id\_apartament AND (id\_adresa <> :NEW.id\_adresa OR numar\_camere <> :NEW.numar\_camere OR etaj <> :NEW.etaj OR suprafata <> :NEW.suprafata OR centrala\_proprie <> :NEW.centrala\_proprie OR pret\_inchiriere <> :NEW.pret\_inchiriere OR tip\_confort <> :NEW.tip\_confort ) ;

ELSIF DELETING THEN

DELETE FROM modbd\_bucuresti.apartament\_bucuresti WHERE id\_apartament = :OLD.id\_apartament;

END IF;

end if;

END;

/

Transaparenta pentru **OLTP\_CONTRACT.**

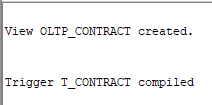
create or replace view oltp\_CONTRACT

as

select \* from modbd\_provincie.contract\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.contract\_bucuresti;



CREATE OR REPLACE TRIGGER t\_contract

INSTEAD OF INSERT OR UPDATE OR DELETE ON oltp\_CONTRACT

FOR EACH ROW

DECLARE

v\_in\_bucuresti NUMBER := 0;

BEGIN

select count (\*) into v\_in\_bucuresti from modbd\_bucuresti.apartament\_bucuresti ap

join modbd\_bucuresti.adresa\_bucuresti adr on ap.id\_adresa = adr.id\_adresa

join modbd\_bucuresti.localitate\_bucuresti loc on adr.id\_localitate = loc.id\_localitate

join modbd\_bucuresti.judet\_bucuresti jud on loc.id\_judet = jud.id\_judet

where ap.id\_apartament = :new.id\_apartament AND jud.nume = 'Bucuresti';

if v\_in\_bucuresti = 0 then

IF INSERTING THEN

BEGIN

INSERT INTO contract\_provincie@bd\_provincie VALUES (PROVINCIE\_CONTRACT\_SEQ.NEXTVAL@bd\_provincie, :NEW.id\_chirias, :NEW.id\_apartament, :NEW.id\_agent, :NEW.data\_inceput, :NEW.data\_final, :NEW.ziua\_scadenta, :NEW.pret\_inchiriere, :NEW.valoare\_estimata, :NEW.incasari);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE contract\_provincie@bd\_provincie SET id\_chirias = :NEW.id\_chirias, id\_apartament = :NEW.id\_apartament, id\_agent = :NEW.id\_agent, data\_inceput = :NEW.data\_inceput, data\_final = :NEW.data\_final, ziua\_scadenta = :NEW.ziua\_scadenta, pret\_inchiriere = :NEW.pret\_inchiriere, valoare\_estimata = :NEW.valoare\_estimata, incasari = :NEW.incasari

WHERE id\_contract = :NEW.id\_contract AND (id\_chirias <> :NEW.id\_chirias OR id\_apartament <> :NEW.id\_apartament OR id\_agent <> :NEW.id\_agent OR data\_inceput <> :NEW.data\_inceput OR data\_final <> :NEW.data\_final OR ziua\_scadenta <> :NEW.ziua\_scadenta OR pret\_inchiriere <> :NEW.pret\_inchiriere OR valoare\_estimata <> :NEW.valoare\_estimata OR incasari <> :NEW.incasari ) ;

ELSIF DELETING THEN

DELETE FROM contract\_provincie@bd\_provincie WHERE id\_contract = :OLD.id\_contract;

END IF;

else

IF INSERTING THEN

BEGIN

INSERT INTO modbd\_bucuresti.contract\_bucuresti VALUES (modbd\_bucuresti.BUCURESTI\_CONTRACT\_SEQ.NEXTVAL, :NEW.id\_chirias, :NEW.id\_apartament, :NEW.id\_agent, :NEW.data\_inceput, :NEW.data\_final, :NEW.ziua\_scadenta, :NEW.pret\_inchiriere, :NEW.valoare\_estimata, :NEW.incasari);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE modbd\_bucuresti.contract\_bucuresti SET id\_chirias = :NEW.id\_chirias, id\_apartament = :NEW.id\_apartament, id\_agent = :NEW.id\_agent, data\_inceput = :NEW.data\_inceput, data\_final = :NEW.data\_final, ziua\_scadenta = :NEW.ziua\_scadenta, pret\_inchiriere = :NEW.pret\_inchiriere, valoare\_estimata = :NEW.valoare\_estimata, incasari = :NEW.incasari

WHERE id\_contract = :NEW.id\_contract AND (id\_chirias <> :NEW.id\_chirias OR id\_apartament <> :NEW.id\_apartament OR id\_agent <> :NEW.id\_agent OR data\_inceput <> :NEW.data\_inceput OR data\_final <> :NEW.data\_final OR ziua\_scadenta <> :NEW.ziua\_scadenta OR pret\_inchiriere <> :NEW.pret\_inchiriere OR valoare\_estimata <> :NEW.valoare\_estimata OR incasari <> :NEW.incasari ) ;

ELSIF DELETING THEN

DELETE FROM modbd\_bucuresti.contract\_bucuresti WHERE id\_contract = :OLD.id\_contract;

END IF;

end if;

END;

/

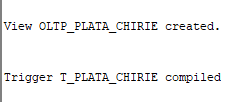
Transaparenta pentru **OLTP\_PLATA\_CHIRIE.**

create or replace view oltp\_plata\_chirie

as

select \* from modbd\_provincie.plata\_chirie\_provincie@bd\_provincie

union all

select \* from modbd\_bucuresti.plata\_chirie\_bucuresti;

CREATE OR REPLACE TRIGGER t\_plata\_chirie

INSTEAD OF INSERT OR UPDATE OR DELETE ON oltp\_plata\_chirie

FOR EACH ROW

DECLARE

v\_in\_bucuresti NUMBER := 0;

BEGIN

select count (\*) into v\_in\_bucuresti from modbd\_bucuresti.contract\_bucuresti c

join modbd\_bucuresti.apartament\_bucuresti ap on c.id\_apartament = ap.id\_apartament

join modbd\_bucuresti.adresa\_bucuresti adr on ap.id\_adresa = adr.id\_adresa

join modbd\_bucuresti.localitate\_bucuresti loc on adr.id\_localitate = loc.id\_localitate

join modbd\_bucuresti.judet\_bucuresti jud on loc.id\_judet = jud.id\_judet

where c.id\_contract = :new.id\_contract AND jud.nume = 'Bucuresti';

if v\_in\_bucuresti = 0 then

IF INSERTING THEN

BEGIN

INSERT INTO plata\_chirie\_provincie@bd\_provincie VALUES (PROVINCIE\_PLATA\_CHIRIE\_SEQ.NEXTVAL@bd\_provincie, :NEW.id\_contract, :NEW.luna, :NEW.an, :NEW.suma, :NEW.data\_efectuarii, :NEW.nr\_zile\_intarziere);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE plata\_chirie\_provincie@bd\_provincie SET id\_contract = :NEW.id\_contract, luna = :NEW.luna, an = :NEW.an, suma = :NEW.suma, data\_efectuarii = :NEW.data\_efectuarii, nr\_zile\_intarziere = :NEW.nr\_zile\_intarziere

WHERE id\_plata = :NEW.id\_plata AND (id\_contract <> :NEW.id\_contract OR luna <> :NEW.luna OR an <> :NEW.an OR suma <> :NEW.suma OR data\_efectuarii <> :NEW.data\_efectuarii OR nr\_zile\_intarziere <> :NEW.nr\_zile\_intarziere) ;

ELSIF DELETING THEN

DELETE FROM plata\_chirie\_provincie@bd\_provincie WHERE id\_plata = :OLD.id\_plata;

END IF;

else

IF INSERTING THEN

BEGIN

INSERT INTO modbd\_bucuresti.plata\_chirie\_bucuresti VALUES (modbd\_bucuresti.BUCURESTI\_PLATA\_CHIRIE\_SEQ.NEXTVAL, :NEW.id\_contract, :NEW.luna, :NEW.an, :NEW.suma, :NEW.data\_efectuarii, :NEW.nr\_zile\_intarziere);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE modbd\_bucuresti.plata\_chirie\_bucuresti SET id\_contract = :NEW.id\_contract, luna = :NEW.luna, an = :NEW.an, suma = :NEW.suma, data\_efectuarii = :NEW.data\_efectuarii, nr\_zile\_intarziere = :NEW.nr\_zile\_intarziere

WHERE id\_plata = :NEW.id\_plata AND (id\_contract <> :NEW.id\_contract OR luna <> :NEW.luna OR an <> :NEW.an OR suma <> :NEW.suma OR data\_efectuarii <> :NEW.data\_efectuarii OR nr\_zile\_intarziere <> :NEW.nr\_zile\_intarziere) ;

ELSIF DELETING THEN

DELETE FROM modbd\_bucuresti.plata\_chirie\_bucuresti WHERE id\_plata = :OLD.id\_plata;

END IF;

end if;

END;

/

# Asigurarea sincronizării datelor pentru relațiile replicate

Relatia **OLTP\_CHIRIAS** a fost replicata, asta inseamna ca trebuie sa avem tot tabelul in ambele baze de date, atat in BUCURESTI cat si in PROVINCIE.

Au fost definit urmatori triggeri:

**Provincie**

CREATE OR REPLACE TRIGGER trig\_rep\_chirias

AFTER INSERT OR UPDATE OR DELETE ON chirias

FOR EACH ROW

DECLARE

v\_dummy NUMBER;

BEGIN

IF INSERTING THEN

BEGIN

INSERT INTO CHIRIAS@bd\_bucuresti VALUES (:NEW.id\_chirias, :NEW.prenume, :NEW.nume, :NEW.telefon, :NEW.email, :NEW.sex, :NEW.data\_nastere, :NEW.starea\_civila);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE CHIRIAS@bd\_bucuresti SET prenume = :NEW.prenume, nume = :NEW.nume, telefon = :NEW.telefon, email = :NEW.email, sex = :NEW.sex, data\_nastere = :NEW.data\_nastere, starea\_civila = :NEW.starea\_civila

WHERE id\_chirias = :NEW.id\_chirias AND (prenume <> :NEW.prenume OR nume <> :NEW.nume OR telefon <> :NEW.telefon OR email <> :NEW.email OR sex <> :NEW.sex OR data\_nastere <> :NEW.data\_nastere OR starea\_civila <> :NEW.starea\_civila );

ELSIF DELETING THEN

DELETE FROM CHIRIAS@bd\_bucuresti WHERE id\_chirias = :OLD.id\_chirias;

END IF;

END;

/

**Bucuresti**

CREATE OR REPLACE TRIGGER trig\_rep\_chirias\_bucuresti

AFTER INSERT OR UPDATE OR DELETE ON chirias

FOR EACH ROW

DECLARE

 v\_dummy NUMBER;

BEGIN

IF INSERTING THEN

BEGIN

INSERT INTO CHIRIAS@bd\_provincie VALUES (:NEW.id\_chirias, :NEW.prenume, :NEW.nume, :NEW.telefon, :NEW.email, :NEW.sex, :NEW.data\_nastere, :NEW.starea\_civila);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

NULL; -- Înregistrarea există deja, nu se face nimic

END;

ELSIF UPDATING THEN

UPDATE CHIRIAS@bd\_provincie SET prenume = :NEW.prenume, nume = :NEW.nume, telefon = :NEW.telefon, email = :NEW.email, sex = :NEW.sex, data\_nastere = :NEW.data\_nastere, starea\_civila = :NEW.starea\_civila

WHERE id\_chirias = :NEW.id\_chirias AND (prenume <> :NEW.prenume OR nume <> :NEW.nume OR telefon <> :NEW.telefon OR email <> :NEW.email OR sex <> :NEW.sex OR data\_nastere <> :NEW.data\_nastere OR starea\_civila <> :NEW.starea\_civila ) ;

ELSIF DELETING THEN

DELETE FROM CHIRIAS@bd\_provincie WHERE id\_chirias = :OLD.id\_chirias;

END IF;

END;

/

# Asigurarea constrângerilor de integritate

Pentru a pastra constrangerile de integritate pe adresa apartamentului am definit doi triggeri before insert in fiecare baza de date locala.

**Bucuresti**

create or replace trigger trigger\_apartament\_adr\_unique

before insert or update on ADRESA\_BUCURESTI

for each row

declare

nr number(1);

begin

select count(\*) into nr

from ADRESA\_PROVINCIE@bd\_provincie

where STRADA = :NEW.STRADA AND NUMAR = :NEW.NUMAR AND BLOC = :NEW.BLOC AND SCARA = :NEW.SCARA AND NUMAR\_APARTAMENT = :NEW.NUMAR\_APARTAMENT;

if (nr<>0) then

raise\_application\_error (-20001,'Constangere de unicitate pe adresa

incalcata.');

end if;

end;

/

**Provincie**

create or replace trigger trigger\_apartament\_adr\_unique

before insert or update on ADRESA\_PROVINCIE

for each row

declare

nr number(1);

begin

select count(\*) into nr

from ADRESA\_BUCURESTI@bd\_bucuresti

where STRADA = :NEW.STRADA AND NUMAR = :NEW.NUMAR AND BLOC = :NEW.BLOC AND SCARA = :NEW.SCARA AND NUMAR\_APARTAMENT = :NEW.NUMAR\_APARTAMENT;

if (nr<>0) then

raise\_application\_error (-20001,'Constangere de unicitate pe adresa incalcata.');

end if;

end;

/