Academia de Studii Economice
Facultatea de Cibernetică, Statistică și Informatică Economică

PROIECT BAZE DE DATE

Lista completă a localităților din România și a facultăților prezente pe teritoriul acesteia

Anul: 2 ID CSIE – Informatică economică
Grupa: 1114
Numele si Prenumele: Richter Cristina

București 2018

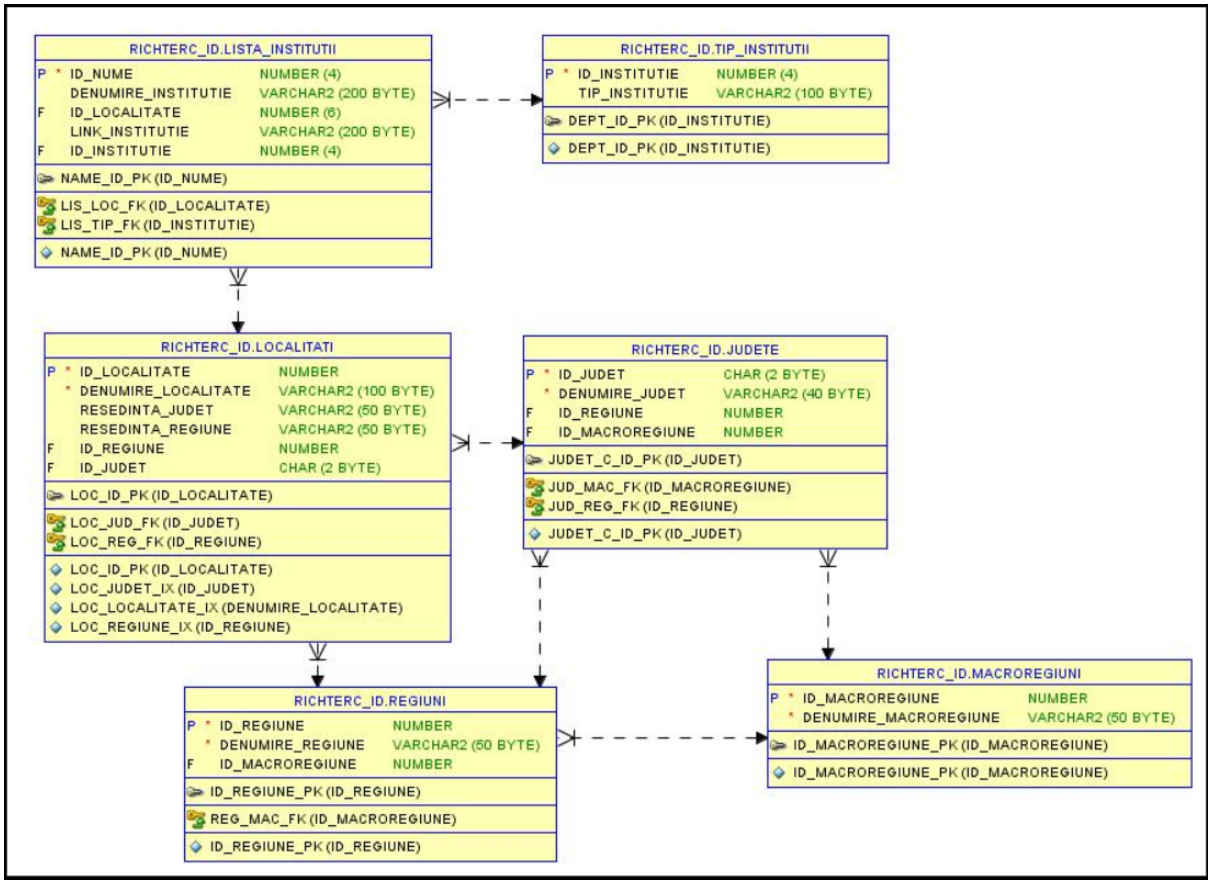


Fig. 1. schema bazei de date proiectata;

Mai jos adaug urmatoarele fisiere sql

1) Creare tabele: „Proiect_Richter_CREATE_TABLES”
2) Stergere tabele daca au fost deja inserate ulterior: „Proiect_Richter_DROP_TABLES”
3) Insert-urile generate pentru a introduce date in tabele „Proiect_Richter_INSERT_DATA”
4) Toate select-urile, update-urile, delete-urile, insert-urile, create-urile, etc folosite in acest proiect.


Proiect_Richter_INSERT_DATA.sql


Proiect_Richter_exemplificari.sql


Proiect_Richter_DROP_TABLES.sql


Proiect_Richter_CREATE_TABLES.sql

Aceste informatii sunt prezente in continuare in documentul de mai jos.

Cuprins

Comenzi LDD (Data Definition Language – Limbaj de Definire a Datelor)

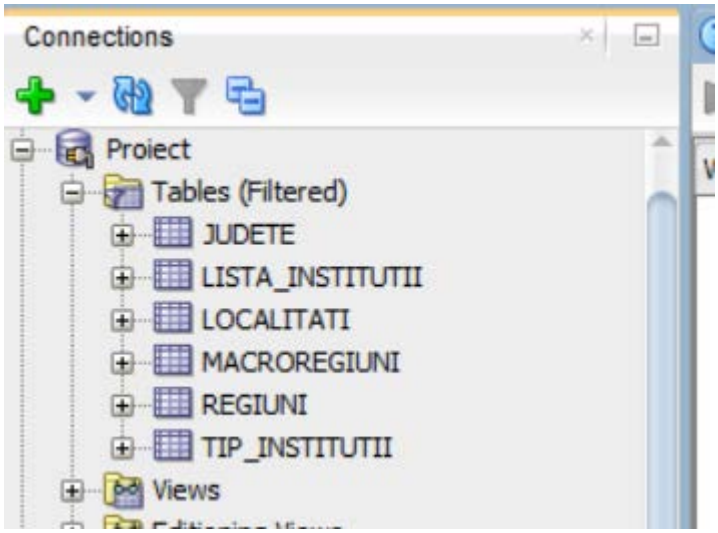
Comanda	Scop
CREATE	Creaza un obiect nou: TABLE, INDEX, CLUSTER, TABLESPACE, SEQUENCE, VIEW, MATERIALIZED VIEW, USER, ROLE, PROCEDURE, FUNCTION, TRIGGER
ALTER	Modifica o parte dintre proprietatile unui obiect
DROP	Sterge un obiect din baza de date

```
prompt
prompt Creating table MACROREGIUNI
prompt ======================
prompt
create table MACROREGIUNI
(
  ID_MACROREGIUNE       NUMBER,
  DENUMIRE_MACROREGIUNE VARCHAR2(50)
)
;
alter table MACROREGIUNI
  add constraint ID_MACROREGIUNE_PK primary key (ID_MACROREGIUNE);
alter table MACROREGIUNI
  add constraint DEN_MACROREGIUNE_NN
  check ("DENUMIRE_MACROREGIUNE" IS NOT NULL);

prompt
prompt Creating table REGIUNI
prompt ======================
prompt
create table REGIUNI
(
  ID_REGIUNE       NUMBER,
  DENUMIRE_REGIUNE VARCHAR2(50),
  ID_MACROREGIUNE  NUMBER
)
;
alter table REGIUNI
  add constraint ID_REGIUNE_PK primary key (ID_REGIUNE);

alter table REGIUNI
  add constraint REG_MAC_FK foreign key (ID_MACROREGIUNE)
  references MACROREGIUNI (ID_MACROREGIUNE);

alter table REGIUNI
  add constraint DEN_REGIUNE_NN
  check ("DENUMIRE_REGIUNE" IS NOT NULL);

prompt
prompt Creating table JUDETE
prompt ====================
prompt
create table JUDETE
(
```

```
  ID_JUDET       CHAR(2),
  DENUMIRE_JUDET VARCHAR2(40),
  ID_REGIUNE     NUMBER,
  constraint JUDET_C_ID_PK primary key (ID_JUDET)
);

alter table JUDETE
    add ID_MACROREGIUNE NUMBER;

alter table JUDETE
  add constraint JUD_REG_FK foreign key (ID_REGIUNE)
  references REGIUNI (ID_REGIUNE);

alter table JUDETE
  add constraint JUD_MAC_FK foreign key (ID_MACROREGIUNE)
  references MACROREGIUNI (ID_MACROREGIUNE);

alter table JUDETE
  add constraint DEN_JUDET_NN
  check ("DENUMIRE_JUDET" IS NOT NULL);

prompt
prompt Creating table LOCALITATI
prompt ======================
prompt
create table LOCALITATI
(
  ID_LOCALITATE         NUMBER not null,
  DENUMIRE_LOCALITATE   VARCHAR2(100),
  RESEDINTA_JUDET       VARCHAR2(50),
  RESEDINTA_REGIUNE     VARCHAR2(50),
  ID_REGIUNE            NUMBER,
  ID_JUDET              CHAR(2)
)
;
alter table LOCALITATI
  add constraint LOC_ID_PK primary key (ID_LOCALITATE);

alter table LOCALITATI
  add constraint LOC_JUD_FK foreign key (ID_JUDET)
  references JUDETE (ID_JUDET);

alter table LOCALITATI
  add constraint LOC_REG_FK foreign key (ID_REGIUNE)
  references REGIUNI (ID_REGIUNE);

alter table LOCALITATI
  add constraint LOC_LOCALITATE_NN
  check ("DENUMIRE_LOCALITATE" IS NOT NULL);

create index LOC_LOCALITATE_IX on LOCALITATI (DENUMIRE_LOCALITATE);
create index LOC_JUDET_IX on LOCALITATI (ID_JUDET);
create index LOC_REGIUNE_IX on LOCALITATI (ID_REGIUNE);

prompt
prompt Creating table TIP_INSTITUTII
prompt ============================
prompt
create table TIP_INSTITUTII
(
  ID_INSTITUTIE       NUMBER(4) not null,
```

```
  TIP_INSTITUTIE      VARCHAR2(100)
)
;
alter table TIP_INSTITUTII
  add constraint DEPT_ID_PK primary key (ID_INSTITUTIE);

prompt
prompt Creating table LISTA_INSTITUTII
prompt ======================
prompt
create table LISTA_INSTITUTII
(
  ID_NUME               NUMBER(4) not null,
  DENUMIRE_INSTITUTIE   VARCHAR2(200),
  ID_LOCALITATE         NUMBER(6),
  LINK_INSTITUTIE       VARCHAR2(200)
)
;
alter table LISTA_INSTITUTII
  add constraint NAME_ID_PK primary key (ID_NUME);

alter table LISTA_INSTITUTII
  add constraint LIS_LOC_FK foreign key (ID_LOCALITATE)
  references LOCALITATI (ID_LOCALITATE);

alter table LISTA_INSTITUTII
  add ID_INSTITUTIE NUMBER(4);

alter table LISTA_INSTITUTII
  add constraint LIS_TIP_FK foreign key (ID_INSTITUTIE)
  references TIP_INSTITUTII (ID_INSTITUTIE);




prompt Disabling triggers for JUDETE...
alter table JUDETE disable all triggers;

prompt Disabling triggers for LISTA_INSTITUTII...
alter table LISTA_INSTITUTII disable all triggers;

prompt Disabling triggers for LOCALITATI...
alter table LOCALITATI disable all triggers;

prompt Disabling triggers for MACROREGIUNI...
alter table MACROREGIUNI disable all triggers;

prompt Disabling triggers for REGIUNI...
alter table REGIUNI disable all triggers;

prompt Disabling triggers for TIP_INSTITUTII...
alter table TIP_INSTITUTII disable all triggers;

--SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME IN
('JUDETE','LISTA_INSTITUTII','LOCALITATI','MACROREGIUNI','REGIUNI','TIP_INS
TITUTII')
--AND CONSTRAINT_TYPE = 'R'
--;
```

```
prompt Disabling foreign key constraints for JUDETE...
alter table JUDETE disable constraint JUD_REG_FK;

prompt Disabling foreign key constraints for LISTA_INSTITUTII...
alter table LISTA_INSTITUTII disable constraint LIS_LOC_FK;
alter table LISTA_INSTITUTII disable constraint LIS_TIP_FK;

prompt Disabling foreign key constraints for LOCALITATI...
alter table LOCALITATI disable constraint LOC_JUD_FK;
alter table LOCALITATI disable constraint LOC_REG_FK;

prompt Disabling foreign key constraints for REGIUNI...
alter table REGIUNI disable constraint REG_MAC_FK;

COMMIT;

DROP TABLE LISTA_INSTITUTII;
DROP TABLE TIP_INSTITUTII;
DROP TABLE LOCALITATI;
DROP TABLE JUDETE;
DROP TABLE REGIUNI;
DROP TABLE MACROREGIUNI;
```



Start Page | *Proiect* | JUDETE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_JUDET	CHAR(2 BYTE)	No	(null)	1	(null)
2	DENUMIRE_JUDET	VARCHAR2(40 BYTE)	Yes	(null)	2	(null)
3	ID_REGIUNE	NUMBER	Yes	(null)	3	(null)
4	ID_MACROREGIUNE	NUMBER	Yes	(null)	4	(null)

Start Page | Proiect | LISTA_INSTITUTII

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_NUME	NUMBER(4,0)	No	(null)	1	(null)
2	DENUMIRE_INSTITUTIE	VARCHAR2(200 BYTE)	Yes	(null)	2	(null)
3	ID_LOCALITATE	NUMBER(6,0)	Yes	(null)	3	(null)
4	LINK_INSTITUTIE	VARCHAR2(200 BYTE)	Yes	(null)	4	(null)
5	ID_INSTITUTIE	NUMBER(4,0)	Yes	(null)	5	(null)

Start Page | Proiect | LOCALITATI

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_LOCALITATE	NUMBER	No	(null)	1	(null)
2	DENUMIRE_LOCALITATE	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	RESEDINTA_JUDET	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	RESEDINTA_REGIUNE	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	ID_REGIUNE	NUMBER	Yes	(null)	5	(null)
6	ID_JUDET	CHAR(2 BYTE)	Yes	(null)	6	(null)

Start Page | Proiect | MACROREGIUNI

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

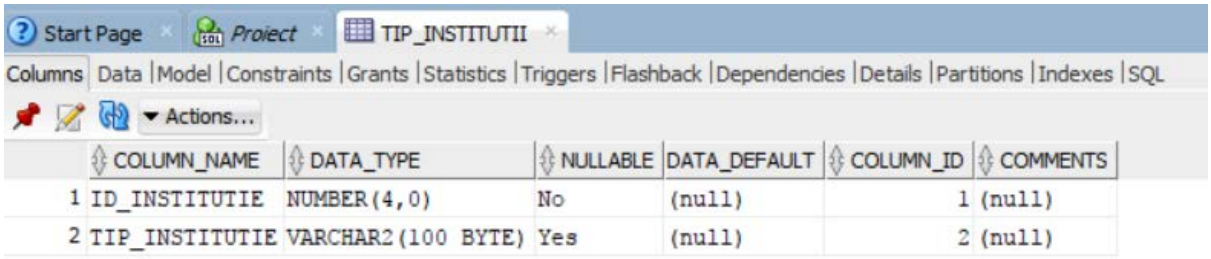
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MACROREGIUNE	NUMBER	No	(null)	1	(null)
2	DENUMIRE_MACROREGIUNE	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)

Start Page | Proiect | REGIUNI

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_REGIUNE	NUMBER	No	(null)	1	(null)
2	DENUMIRE_REGIUNE	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	ID_MACROREGIUNE	NUMBER	Yes	(null)	3	(null)



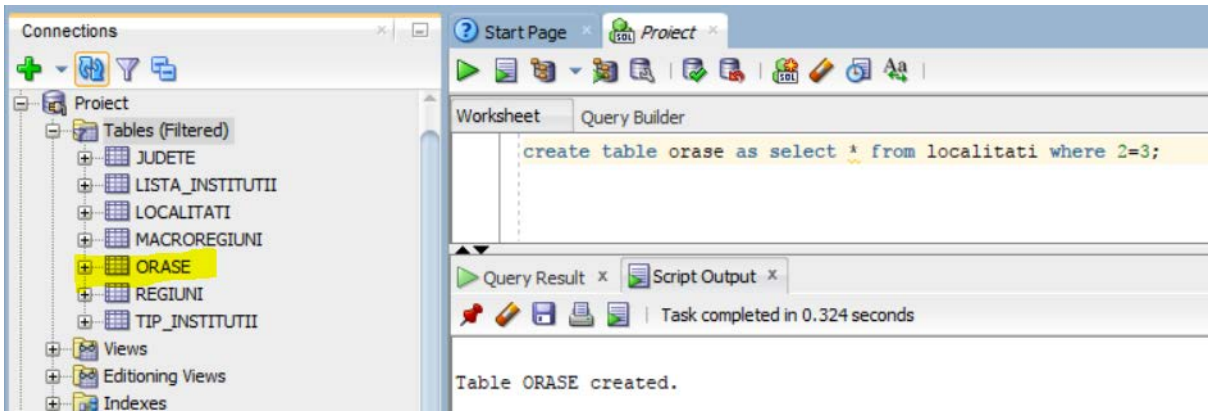
b) Adaugarea inregistrarilor pe baza valorilor din alte tabele:

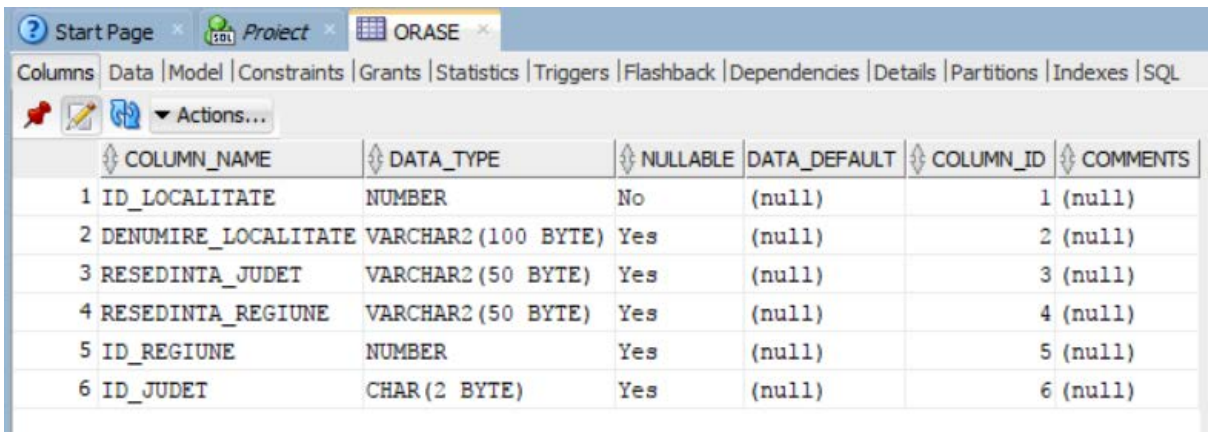
Exemplu 1. Sa creeze tabela **ORASE** pe baza tabelei **LOCALITATI** fara a prelua si inregistrarile (doar structura) si sa se adauge o noua localitate.

```
CREATE TABLE orase AS SELECT * FROM localitati WHERE 2=3;
```

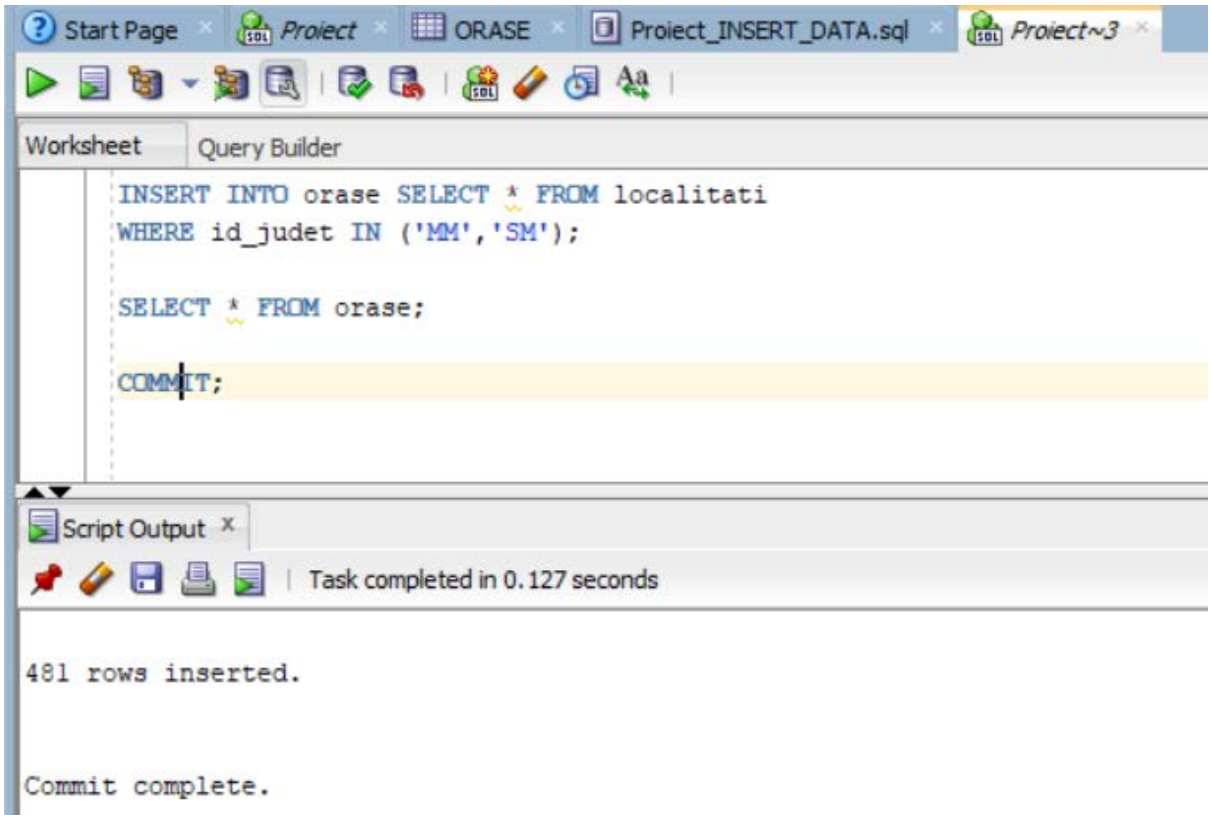
- nu se insereaza date pentru ca conditia 2=3 nu e niciodata adevarata.

```
Insert into ORASE
(ID_LOCALITATE,DENUMIRE_LOCALITATE,RESEDINTA_JUDET,RESEDINTA_REGIUNE,ID_REG
IUNE,ID_JUDET) values (1126,Oras_test',null,null,6,'SJ');
```





Exemplul 2. Sa se adauge in tabela **ORASE** toate localitatile din tabela **LOCALITATI** care apartin judetelor Maramures si Satu Mare. Si sa se finalizeze tranzactia (salvarea modificarii).



Exemplul 4

4.1 Sa se redenumeasca tabela orase cu comune si apoi invers

```
ALTER TABLE orase RENAME TO comune;
```

Sau

```
RENAME comune TO orase;
```

4.2 Sa se adauge coloanele numar_locuitori si numar_muzee in tabela orase

```
ALTER TABLE orase
ADD (numar_muzee NUMBER(2),
numar_locuitori NUMBER(20));
```

4.3 Sa se modifice tipul de date al coloanei numar_locuitori

```
ALTER TABLE orase MODIFY (
    numar_locuitori NUMBER(30)
);
```

4.4 Sa se stearga coloana numar_muzee

```
ALTER TABLE orase DROP COLUMN numar_muzee;
```

4.5 Sa se inactiveze coloana numar_locuitori

```
ALTER TABLE orase SET UNUSED COLUMN numar_locuitori;
```

4.6 Sa se stearga coloanele inactive

ALTER TABLE personal

DROP UNUSED COLUMNS;

4.7 Sa se adauge o restrictie pe coloana varsta

ALTER TABLE personal

ADD CONSTRAINT check_varsta CHECK (varsta>18 and varsta<60);

4.8 Sa se dezactiveze restrictia anterioara

ALTER TABLE personal

DISABLE CONSTRAINT check_varsta;

4.9 Sa se stearga restrictia anterioara

ALTER TABLE personal

DROP CONSTRAINT check_varsta;

```
--CREAREA UNEI TABELE PE BAZA COLOANELOR DIN ALTA TABELA:


--Exemplul 3. Sa se creeze tabela orase_bkup pe baza tabelei localitati dar
care sa contina numai localitatile din anumite judete.

CREATE TABLE orase_bkup
AS
SELECT * FROM localitati
WHERE id_judet in ('MM','SM','BV');


--create table orase_bkup;

INSERT INTO orase_bkup SELECT * FROM orase
WHERE id_judet IN ('MM', 'SM', 'SB');

SELECT * FROM salariati;

COMMIT;
--Exemplul 3. Sa se creeze tabela orase pe baza tabelei localitati;
CREATE TABLE orase
AS
SELECT * FROM localitati;

drop table orase_bkup;
--Exemplul 3. Sa se creeze tabela orase pe baza tabelei localitati
--si care va contine doar o parte din coloanele tabelei initiale (codagent,
numeagent, functia, codfirma)
CREATE TABLE orase_bkup
AS
```

```sql
SELECT denumire_localitate, id_localitate FROM localitati;

--MODIFICAREA STRUCTURII TABELELOR - COMANDA ALTER TABLE

--Comanda ALTER TABLE permite:
--- Modificarea structurii unei tabele avand urmatoarele optiuni:
--Adaugare coloana  ADD nume_coloana tip_data
--Modificare coloana    MODIFY nume_coloana tip_nou_data
--Stergere coloana  DROP COLUMN nume_coloana
--Inactivare coloana in vederea stergerii ulterioare    SET UNUSED
--
--- Modificarea restrictiilor de integritate avand urmatoarele optiuni:
--Adaugare restrictie   ADD CONSTRAINT nume_restrictie tip_restrictie
--Modificare restrictie MODIFY CONSTRAINT nume_restrictie
tip_nou_restrictie
--Stergere restrictie   DROP CONSTRAINT nume_restrictie
--Dezactivare/activare restrictie   DISABLE/ENABLE CONSTRAINT
nume_restrictie

--- Redenumirea tabelei: RENAME
ALTER TABLE orase_bkup RENAME TO orase_backup;
--sau
RENAME orase_backup TO orase_bkup;

--4.2 Sa se adauge coloanele email si data_infintare in tabela
tip_institutii
ALTER TABLE tip_institutii
ADD (email VARCHAR2(10),
data_infintare date);

--4.3  Sa se modifice tipul de date al coloanei email
ALTER TABLE tip_institutii
MODIFY (email VARCHAR2(30));

--4.4 Sa se stearga coloana email
ALTER TABLE tip_institutii
DROP COLUMN email;

--4.5 Sa se inactiveze coloana functia
ALTER TABLE tip_institutii
SET UNUSED COLUMN data_infintare;

--4.6 Sa se stearga coloanele inactive
ALTER TABLE tip_institutii
DROP UNUSED COLUMNS;

--4.7 Sa se adauge o restrictie pe coloana varsta
ALTER TABLE tip_institutii
ADD CONSTRAINT check_varsta CHECK (varsta>18 and varsta<60);

--4.8 Sa se dezactiveze restrictia anterioara
ALTER TABLE tip_institutii
DISABLE CONSTRAINT check_varsta;

--4.9 Sa se stearga restrictia anterioara
ALTER TABLE tip_institutii
DROP CONSTRAINT check_varsta;


--STERGEREA TABELELOR – COMENZILE DROP TABLE  SI TRUNCATE TABLE
```

```
--Comanda permite stergerea unei tabele [inclusiv restrictiile acesteia] cu
posibilitate de recuperare:
DROP TABLE orase_bkup CASCADE CONSTRAINTS;
FLASHBACK TABLE orase_bkup TO BEFORE DROP;

--Sterge definitiv tabela fara posibilitate de recuperare
DROP TABLE orase_bkup PURGE;

--Comanda TRUNCATE TABLE sterge inregistrarile unei tabele si elibereaza
spatiul alocat acestora

--Exemplul 6. Sa se stearga inregistrarile tabelei personal
TRUNCATE TABLE orase_bkup;

--VIZUALIZAREA OBIECTELOR CE APARTIN UNUI ANUMIT UTILIZATOR:
--Din dictionarul bazei de date se pot vizualiza o serie de informatii
referitoare la obiectele utilizatorului curent

--Exemplul 7. Sa se vizualizeze toate tabelele utilizatorului curent
SELECT * FROM USER_TABLES;

--Exemplul 8. Sa se vizualizeze denumirea tabelelor, restrictiile si tipul
acestora pentru utilizatorul curent
SELECT TABLE_NAME, CONSTRAINT_TYPE, CONSTRAINT_NAME
FROM USER_CONSTRAINTS;


--Exemplul 4
--4.1 Sa se redenumeasca tabela orase cu comune si apoi invers
ALTER TABLE orase RENAME TO comune;

--Sau

RENAME comune TO orase;

--4.2 Sa se adauge coloanele numar_locuitori si numar_muzee in tabela orase
ALTER TABLE orase
ADD (numar_muzee NUMBER(2),
numar_locuitori NUMBER(20));

--4.3  Sa se modifice tipul de date al coloanei numar_locuitori
ALTER TABLE orase MODIFY (
    numar_locuitori NUMBER(30)
);

--4.4 Sa se stearga coloana numar_muzee
ALTER TABLE orase DROP COLUMN numar_muzee;

--4.5 Sa se inactiveze coloana numar_locuitori
ALTER TABLE orase SET UNUSED COLUMN numar_locuitori;

--4.6 Sa se stearga coloanele inactive
ALTER TABLE orase
DROP UNUSED COLUMNS;

--4.7 Sa se adauge o restrictie pe coloana varsta
ALTER TABLE orase
ADD CONSTRAINT check_id_judet CHECK (id_judet is not null);

--4.8 Sa se dezactiveze restrictia anterioara
ALTER TABLE orase
```

```
DISABLE CONSTRAINT check_id_judet;

--4.9 Sa se stearga restrictia anterioara
ALTER TABLE orase
DROP CONSTRAINT check_id_judet;

--Sterge definitiv tabela fara posibilitate de recuperare
--DROP TABLE orase PURGE;

--Exemplul 5.
--5.1 Sa se stearga tabela orase
DROP TABLE orase CASCADE CONSTRAINTS;
--5.2 Sa se recupereze tabela orase
FLASHBACK TABLE orase TO BEFORE DROP;

--Comanda TRUNCATE TABLE sterge inregistrarile unei tabele si elibereaza
spatiul alocat acestora

--Exemplul 6. Sa se stearga inregistrarile tabelei orase
TRUNCATE TABLE orase;

--VIZUALIZAREA OBIECTELOR CE APARTIN UNUI ANUMIT UTILIZATOR:
--Din dictionarul bazei de date se pot vizualiza o serie de informatii
referitoare la obiectele utilizatorului curent

--Exemplul 7. Sa se vizualizeze toate tabelele utilizatorului curent si sa
se ordoneze descrescator in functie de num_rows.
SELECT NUM_ROWS, U.* FROM USER_TABLES U
ORDER BY U.NUM_ROWS DESC;

--Exemplul 8. Sa se vizualizeze denumirea tabelelor, restrictiile si tipul
acestora pentru utilizatorul curent
SELECT TABLE_NAME, CONSTRAINT_TYPE, CONSTRAINT_NAME
FROM USER_CONSTRAINTS;


--ACTUALIZAREA TABELELOR - COMENZI LMD
-- (Data Manipulation Language – Limbaj de Manipulare a Datelor)


--b) Adaugarea inregistrarilor pe baza valorilor din alte tabele:
/*
--Exemplul 2. Sa se adauge in tabela orase_bkup toti angajatii din tabela
--orase_bkup care lucreaza din judetele ('MM', 'SM', 'SB').
--Si sa se finalizeze tranzactia (salvarea modificarii).
*/


--c) Utilizarea variabilelor de substitutie pentru adaugarea
inregistrarilor pe baza valorilor introduse de utilizator de la tastatura:
--INSERT INTO nume_tabela (lista coloane) VALUES (&valoare_coloana1,
&valoare_coloana2,...);

--Exemplul 3. Sa se adauge in tabela salariati un angajat ale carui date
sunt introduse de utilizator de la tastatura
INSERT INTO judete  (id_judet, denumire_judet, id_regiune, id_macroregiune)
VALUES ('&id_judet','&denumire_judet', '&id_regiune',  '&id_macroregiune');


--STERGEREA DATELOR – COMANDA DELETE
```

```
DELETE FROM orase
WHERE id_judet ='SM';


--Exemplul 9. Sa se sterga toti angajatii din tabela salariati. Sa se
anuleze tranzactia.
DELETE FROM orase_bkup;

SELECT * FROM orase_bkup;

ROLLBACK;

SELECT * FROM orase_bkup;

--ACTUALIZAREA DATELOR CU COMANDA MERGE


--Exemplul 10. Sa se actualizeze tabela orase_bkup astfel incat toate
orasele din
--tabela orase_bkup sa aiba localitatile la fel cu dele din tabela orase,
iar pentru
--cei care nu sunt in tabela orase_bkup sa se adauge valorile coloanelor
(id_localitate, denumire_localitate)
--din tabela sursa orase. Sa se numere inregistrarile din cele doua tabele
si sa se explice diferenta. Sa se finalizeze tranzactia.
```

Inserare date in tabele

Mai jos sunt doar exemple de insert-uri. Pentru ca am introdus foarte multe date nu le pot adauga pe toate aici. Lista completa de insert-uri se gaseste in fisierul sql, aflat la inceputul documentului cu numele „Proiect_Richter_INSERT_DATA.sql”

```
--Row 1
INSERT INTO MACROREGIUNI (ID_MACROREGIUNE, DENUMIRE_MACROREGIUNE) VALUES
(1,'MACROREGIUNEA UNU');
--Row 2
INSERT INTO MACROREGIUNI (ID_MACROREGIUNE, DENUMIRE_MACROREGIUNE) VALUES
(2,'MACROREGIUNEA DOI');
--Row 3
INSERT INTO MACROREGIUNI (ID_MACROREGIUNE, DENUMIRE_MACROREGIUNE) VALUES
(3,'MACROREGIUNEA TREI');
--Row 4
INSERT INTO MACROREGIUNI (ID_MACROREGIUNE, DENUMIRE_MACROREGIUNE) VALUES
(4,'MACROREGIUNEA PATRU');

--SET DEFINE OFF

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE, ID_MACROREGIUNE)
VALUES (1, 'Nord Est', 2);

INSERT INTO REGIUNI (ID_REGIUNE, DENUMIRE_REGIUNE, ID_MACROREGIUNE)
VALUES (2, 'Sud Est', 2);
```

```
Insert into RICHTERC_ID.LOCALITATI
(ID_LOCALITATE,DENUMIRE_LOCALITATE,RESEDINTA_JUDET,RESEDINTA_REGIUNE,ID_REG
IUNE,ID_JUDET) values (1126,'Bizusa Bai',null,null,6,'SJ');
Insert into RICHTERC_ID.LOCALITATI
(ID_LOCALITATE,DENUMIRE_LOCALITATE,RESEDINTA_JUDET,RESEDINTA_REGIUNE,ID_REG
IUNE,ID_JUDET) values (1127,'Blaga',null,null,1,'BC');
Insert into RICHTERC_ID.LOCALITATI
(ID_LOCALITATE,DENUMIRE_LOCALITATE,RESEDINTA_JUDET,RESEDINTA_REGIUNE,ID_REG
IUNE,ID_JUDET) values (1128,'Blaga',null,null,1,'IS');


INSERT INTO LISTA_INSTITUTII (ID_NUME, ID_INSTITUTIE, DENUMIRE_INSTITUTIE,
ID_LOCALITATE, LINK_INSTITUTIE)
VALUES (91, 4, 'INSTITUTUL TEOLOGIC ROMANO-CATOLIC FRANCISCAN DIN ROMAN',
10024, 'www.itrcf.ofmconv.ro');

INSERT INTO LISTA_INSTITUTII (ID_NUME, ID_INSTITUTIE, DENUMIRE_INSTITUTIE,
ID_LOCALITATE, LINK_INSTITUTIE)
VALUES (92, 4, 'FUNDATIA "ACADEMIA COMERCIALA" DIN SATU MARE', 10636,
'www.academiacomerciala.ro');

-- Import Data into table LISTA_INSTITUTII from file
C:\Users\Cristina\Desktop\Proiect Baze de date.xlsx . Task successful and
sent to worksheet.
commit;
```

Query-uri

ACTUALIZAREA DATELOR CU COMANDA MERGE

```
--Exemplul 10. Sa se actualizeze tabela orase_bkup astfel incat toate
orasele din
--tabela orase_bkup sa aiba localitatile la fel cu dele din tabela orase,
iar pentru
--cei care nu sunt in tabela orase_bkup sa se adauge valorile coloanelor
(id_localitate, denumire_localitate)
--din tabela sursa orase. Sa se numere inregistrarile din cele doua tabele
si sa se explice diferenta. Sa se finalizeze tranzactia.

MERGE INTO orase_bkup b USING orase o
ON (b.id_localitate = o.id_localitate)
WHEN MATCHED THEN
UPDATE SET b.denumire_localitate=o.denumire_localitate
WHEN NOT MATCHED THEN
INSERT  (id_localitate, denumire_localitate) VALUES (o.id_localitate,
o.denumire_localitate);

SELECT COUNT (*) FROM orase_bkup;
SELECT COUNT (*) FROM orase;

COMMIT;
```

Operatorul ANY si operatorul ALL

```
--ANY Comapara valoarea cu oricare valoare returnata de interogare
--ALL compara valoarea cu fiecare valoare returnata de interogare

SELECT id_localitate, denumire_localitate, resedinta_judet, id_judet
FROM localitati
WHERE id_judet < ANY
 (SELECT id_judet FROM judete
WHERE id_regiune = 1)
AND id_regiune <> 5
ORDER BY denumire_localitate DESC;


Sa se afiseze id_localitate, denumire_localitate, resedinta_judet si
--id_judet pentru localitatile care nu sunt in regiunea 5, ordonare
descrescatoare dupa denumire_localitate
SELECT id_localitate, denumire_localitate, resedinta_judet, id_judet
FROM localitati
WHERE id_judet < ALL
 (SELECT id_judet FROM judete
WHERE id_regiune = 1)
AND id_regiune <> 5
ORDER BY denumire_localitate DESC;
```

--Realizarea jonctiunilor între relatii.

```
/*10.   Sa se selecteze institutiile de invatamant superior de stat (civile
sau
militare) din regiunea vest, afisand si localitate, judet, regiune si
macroregiune.
*/

SELECT
    j.denumire_judet AS "Judet",
    m.denumire_macroregiune "MacroReg",
    l.denumire_localitate AS localitate,
    li.denumire_institutie AS institutie,
    ti.tip_institutie AS TIP,
    r.denumire_regiune AS regiune
FROM
    tip_institutii ti,
    lista_institutii li,
    localitati l,
--    orase o, --inline comment
    judete j,
    regiuni r,
    macroregiuni m
WHERE        ti.id_institutie = li.id_institutie
    AND        l.id_localitate = li.id_localitate
--    AND        l.id_localitate = o.id_localitate
    AND        l.id_judet = j.id_judet
    AND        r.id_regiune = l.id_regiune
    AND        r.id_macroregiune = m.id_macroregiune
    AND        link_institutie IS NOT NULL
    AND        denumire_institutie LIKE upper('universitate%')
    AND        tip_institutie LIKE ( 'Institutii de invatamant superior de
stat%' )
```

```
    AND         r.denumire_regiune LIKE ( '%Vest%' )
ORDER BY li.id_nume ASC;

/*
Arad    MACROREGIUNEA PATRU Arad    UNIVERSITATEA "AUREL VLAICU" DIN ARAD
Institutii de invatamant superior de stat civile    Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA TEHNICA DIN CLUJ -
NAPOCA Institutii de invatamant superior de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE STIINTE AGRICOLE
SI MEDICINA VETERINARA DIN CLUJ-NAPOCA    Institutii de invatamant superior
de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA "BABES-BOLYAI" DIN
CLUJ-NAPOCA    Institutii de invatamant superior de stat civile    Nord
Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE MEDICINA SI
FARMACIE "IULIU HATIEGANU" DIN CLUJ-NAPOCA Institutii de invatamant
superior de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE ARTA SI DESIGN DIN
CLUJ-NAPOCA Institutii de invatamant superior de stat civile    Nord Vest
Dolj    MACROREGIUNEA PATRU Craiova UNIVERSITATEA DIN CRAIOVA   Institutii
de invatamant superior de stat civile    Sud Vest - Oltenia
Dolj    MACROREGIUNEA PATRU Craiova UNIVERSITATEA DE MEDICINA SI FARMACIE
DIN CRAIOVA   Institutii de invatamant superior de stat civile    Sud Vest
- Oltenia
Bihor   MACROREGIUNEA UNU   Oradea  UNIVERSITATEA DIN ORADEA    Institutii
de invatamant superior de stat civile    Nord Vest
Caras-Severin   MACROREGIUNEA PATRU Resita  UNIVERSITATEA "EFTIMIE MURGU"
DIN RESITA    Institutii de invatamant superior de stat civile    Vest
Gorj    MACROREGIUNEA PATRU Targu Jiu   UNIVERSITATEA "CONSTANTIN BRANCUSI"
DIN TARGU JIU   Institutii de invatamant superior de stat civile    Sud
Vest - Oltenia
Timis   MACROREGIUNEA PATRU Timisoara   UNIVERSITATEA "POLITEHNICA"
TIMISOARA   Institutii de invatamant superior de stat civile    Vest
Timis   MACROREGIUNEA PATRU Timisoara   UNIVERSITATEA DE STIINTE AGRICOLE
SI MEDICINA VETERINARA A BANATULUI "REGELE MIHAI I AL ROMANIEI" DIN
TIMISOARA Institutii de invatamant superior de stat civile    Vest
*/
```

BETWEEN;

IS NOT NULL;

LIKE

ORDER BY

b. Jonctiune externa

```
SELECT
    j.denumire_judet AS "Judet",
    m.denumire_macroregiune "MacroReg",
    l.denumire_localitate AS localitate,
    li.denumire_institutie AS institutie,
    ti.tip_institutie AS TIP,
    r.denumire_regiune AS regiune
FROM
    tip_institutii ti,
    lista_institutii li,
    localitati l,
--    orase o, --inline comment
```

```
    judete j,
    regiuni r,
    macroregiuni m
WHERE        ti.id_institutie(+) = li.id_institutie --left join
    AND        l.id_localitate = li.id_localitate(+) --right join
--    AND        l.id_localitate = o.id_localitate
    AND        l.id_judet = j.id_judet
    AND        r.id_regiune = l.id_regiune
    AND        r.id_macroregiune = m.id_macroregiune
    AND        link_institutie IS NOT NULL
    AND        denumire_institutie LIKE upper('universitate%')
    AND        tip_institutie LIKE ( 'Institutii de invatamant superior de
stat%' )
    AND        r.denumire_regiune LIKE ( '%Vest%' )
ORDER BY li.id_nume ASC;

/*
Arad    MACROREGIUNEA PATRU Arad    UNIVERSITATEA "AUREL VLAICU" DIN ARAD
Institutii de invatamant superior de stat civile    Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA TEHNICA DIN CLUJ -
NAPOCA Institutii de invatamant superior de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE STIINTE AGRICOLE
SI MEDICINA VETERINARA DIN CLUJ-NAPOCA    Institutii de invatamant superior
de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA "BABES-BOLYAI" DIN
CLUJ-NAPOCA    Institutii de invatamant superior de stat civile    Nord
Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE MEDICINA SI
FARMACIE "IULIU HATIEGANU" DIN CLUJ-NAPOCA Institutii de invatamant
superior de stat civile    Nord Vest
Cluj    MACROREGIUNEA UNU   Cluj-Napoca UNIVERSITATEA DE ARTA SI DESIGN DIN
CLUJ-NAPOCA Institutii de invatamant superior de stat civile    Nord Vest
Dolj    MACROREGIUNEA PATRU Craiova UNIVERSITATEA DIN CRAIOVA   Institutii
de invatamant superior de stat civile    Sud Vest - Oltenia
Dolj    MACROREGIUNEA PATRU Craiova UNIVERSITATEA DE MEDICINA SI FARMACIE
DIN CRAIOVA   Institutii de invatamant superior de stat civile    Sud Vest
- Oltenia
Bihor   MACROREGIUNEA UNU   Oradea  UNIVERSITATEA DIN ORADEA    Institutii
de invatamant superior de stat civile    Nord Vest
Caras-Severin   MACROREGIUNEA PATRU Resita  UNIVERSITATEA "EFTIMIE MURGU"
DIN RESITA    Institutii de invatamant superior de stat civile    Vest
*/


 --c. Jonctiunea dintre o tabela cu aceeasi tabela => SELF JOIN
 -- + folosire BETWEEN
 -- + folosire RIGHT JOIN & LEFT JOIN
 -- + folosire UPPER, NOT NULL & LIKE
 -- + exemplu concatenare

SELECT
    l.denumire_localitate ||' este in judetul: '||j.denumire_judet AS
"concatenare",
    m.denumire_macroregiune "MacroReg",
    li.denumire_institutie AS institutie,
    ti.tip_institutie AS TIP,
    r.denumire_regiune AS regiune
FROM
    tip_institutii ti,
```

```sql
    lista_institutii li, --self join
    lista_institutii li2, --self join
    localitati l,
    judete j,
    regiuni r,
    macroregiuni m
WHERE        ti.id_institutie(+) = li.id_institutie --left join
    AND        l.id_localitate = li.id_localitate(+) --right join
    AND        li.id_institutie = li2.id_institutie
    AND        l.id_judet = j.id_judet
    AND        r.id_regiune = l.id_regiune
    AND        r.id_macroregiune = m.id_macroregiune
    AND        li.link_institutie IS NOT NULL
    AND        li.denumire_institutie LIKE upper('universitate%')
    AND        ti.tip_institutie LIKE ( 'Institutii de invatamant superior
de stat%' )
--    AND        r.denumire_regiune LIKE ( '%Vest%' )
    AND        l.id_localitate between 1 and 999
ORDER BY li.id_nume ASC;

/*
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
Alba Iulia este in judetul: Alba    MACROREGIUNEA UNU   UNIVERSITATEA "1
DECEMBRIE 1918" DIN ALBA IULIA Institutii de invatamant superior de stat
civile    Centru
*/

--Realizarea interogarilor subordonate (se utilizeaza 2 comenzi SELECT
imbricate)

--15.   Sa se selecteze angajatii care sunt in acelasi departament cu
angajatul Smith.
```

--+ folosire UPPER, LOWER

```sql
SELECT * FROM localitati
WHERE id_judet IN
(SELECT id_judet FROM judete WHERE upper(denumire_judet)like
upper('%mar%'));
```

--Clauza FOR UPDATE

```
-- blocheaza randurile selectate de o interogare in vederea
--actualizarii ulterioare, ceilalti utilizatori nu pot modifica acele
randuri pana la finalizarea tranzactiei;
--FOR UPDATE nu se foloseste cu:
```

--• DISTINCT

--• GROUP BY

--• Functii de grup

```
SELECT a.id_macroregiune, a.denumire_macroregiune, c.denumire_judet
FROM macroregiuni a, judete c
WHERE a.id_macroregiune = c.id_macroregiune
FOR UPDATE;


--UTILIZAREA FUNCTIILOR PREDEFINITE IN INTEROGARI
--•   Functii de tip single-row (sau scalare). O functie single-row
întoarce un singur rezultat pentru fiecare rând al tabelei interogate sau
view
--•   Functii de grup (sau agregate). O functie de grup întoarce un singur
rezultat pentru un grup de rânduri interogate. Functiile de grup pot aparea
în clauza HAVING.
```

--FUNCTII SINGLE-ROW

--Functii de tip caracter Functia LOWER() , UPPER(), INITCAP()

```
--1.    Sa se afiseze cu litere mari denumirea localitatilor din
macroregiunea 4:

SELECT
    l.id_localitate,
    initcap(l.denumire_localitate)
FROM
    localitati l,
    regiuni r,
    macroregiuni m
WHERE        r.id_macroregiune = 4
    AND        l.id_regiune = r.id_regiune
    AND        r.id_macroregiune = m.id_macroregiune;

/*
572 Balcesti
573 Balcesti
577 Baldovin
579 Baldovinesti
581 Baleasa
587 Balesti
595 Balint
601 Balomir
602 Balomireasa
604 Balosani
605 Balosesti
*/
```

--Functia CONCAT() , functia LENGTH() , functia SUBSTR()

```
--5.    S? se afi?eze id_client, numele clientilor concatenat? cu sexul
acestora ?i lungimea prenumelui, nivel_venituri numai pentru clientii cu
venituri in categoria F: 110000 - 129999
SELECT
    id_judet,
    concat(denumire_judet,id_judet),
    length(denumire_judet),
    denumire_judet
FROM
    judete
WHERE
    substr(denumire_judet,1,1) = 'M';

    /*
    MM  MaramuresMM 9   Maramures
    MH  MehedintiMH 9   Mehedinti
    MS  MuresMS 5   Mures
    */
```

--Functii de tip numeric Functia ROUND(), TRUNC()

```
--6.    Sa se afiseze numarul 45,923 rotunjit la doua zecimale si rotunjit
la numar intreg.
--Sa se aplice si functia TRUNC.
 SELECT ROUND(45.923,2), ROUND(45.923,0) FROM DUAL; --45.92 46
 SELECT TRUNC(45.923,2), TRUNC(45.923,0) FROM DUAL; --45.92 45
```

--Functii de tip data calendaristica

--Functia SYSDATE

```
--8.    Afisati data curenta (se selecteaza data din tabela DUAL):

SELECT SYSDATE data_curenta FROM DUAL; --17-JAN-18
SELECT to_char(SYSDATE,'dd/mm/yyyy hh24:mi:ss') data_curenta FROM DUAL; --
17/01/2018 15:42:57
```

--Functiile MONTH_BETWEEN() , ADD_MONTHS() , NEXT_DAY() , LAST_DAY()

```
select
LAST_DAY(sysdate), --ultima zi a lunii
ADD_MONTHS(sysdate,2) --adauga 2 luni de la sysdate
FROM dual; --31-JAN-18  17-MAR-18



--10.   Sa se afiseze comenzile incheiate in luna trecuta:

SELECT nr_comanda, data FROM comenzi
WHERE round(MONTHS_BETWEEN(sysdate, data))=171;
```

--Functia TO_NUMBER

```
-- Converteste sirul de caractere intr-un numar cu un anumit format
--TO_NUMBER(char[, 'format_model'])
```

--Functia EXTRACT()

```
--14.   Sa se afiseze comenzile incheiate in anii 1997 si 1998.
```

```sql
SELECT nr_comanda, data  FROM comenzi
    WHERE EXTRACT(YEAR FROM data) IN (1997, 1998);

--15.   Sa se afiseze comenzile incheiate in lunile iulie si august.

SELECT nr_comanda, data  FROM comenzi
    WHERE EXTRACT(MONTH FROM data) IN (7,8);
```

--Functiile NVL, NVL2, NULLIF, COALESCE

```sql
--16.   Se inmulteste id_ul localitatii cu 10, campul resedinta judet apare
completat cu N daca este null
-- nvl2 afiseaza daca localitatea este resedinta de regiune sau nu
SELECT
    id_localitate * 10,
    denumire_localitate,
    nvl(resedinta_judet,'N'),
    nvl2(resedinta_regiune,'Y','N')
FROM    localitati
WHERE        resedinta_regiune IS NOT NULL
    OR        resedinta_judet IS NOT NULL;

    /*
    11100   Bistrita    Y   N
    840 Alba Iulia  Y   Y
    1260    Alexandria  Y   N
    2570    Arad    Y   N
    4180    Bacau   Y   N
    4900    Baia Mare   Y   N
    19730   Buzau   Y   N
    20350   Calarasi    Y   Y
    29730   Cluj-Napoca Y   Y
    */
/*
```

NULLIF

```sql
18. Sa se afiseze lungimea denumirii orasului si a judetului,
daca acestea sunt egale sa se returneze nul ca rezultat,
iar daca nu sunt egale se va returna lungimea denumirii orasului.

se afiseaza de asemenea daca localitatea este resedinta de regiune, daca nu
este
afiseaza ca este resedinta de judet,
daca nu este nici resedinta judet afiseaza cuvantul "nimic".

*/
SELECT
    l.denumire_localitate,
    length(l.denumire_localitate),
    j.denumire_judet,
    length(j.denumire_judet),
    nullif(
        length(l.denumire_localitate),
        length(j.denumire_judet)
    ) rezultat,
    coalesce(resedinta_regiune,resedinta_judet,'nimic')
FROM
    localitati l,
    judete j
WHERE L.id_judet = j.id_judet
```

```
    ;

/*
Balcani 7   Bacau   5   7   nimic
Balcauti    8   Suceava 7   8   nimic
Balcesti    8   Cluj    4   8   nimic
Balcesti    8   Cluj    4   8   nimic
Balcesti    8   Gorj    4   8   nimic
Balcesti    8   Valcea  6   8   nimic
Balciu  6   Iasi    4   6   nimic
*/


--FUNCTII DE GRUP

--AVG([DISTINCT|ALL] n) – calculeaza media aritmetica a valorilor
--COUNT(* | [DISTINCT|ALL] expr) – intoarce numarul total al valorilor
--MAX([DISTINCT|ALL] expr) – intoarce valoarea maxima MIN([DISTINCT|ALL]
expr) – intoarce valoarea minima
--SUM([DISTINCT|ALL] n) – calculeaza suma valorilor
--
--Se utilizeaza urmatoarele clauze:
--GROUP BY – grupeaza datele in functie de un anumit camp;
--ORDER BY – ordoneaza datele in functie de un anumit camp;
--HAVING – permite stabilirea unor criterii de selectie asupra functiilor
de grup;
--
--Exemple:
--1.    calcule:

AVG

SELECT
    AVG(j.id_macroregiune * j.id_regiune) as medie, --
9.166666666666666666666666666666666666667
    MAX(j.id_macroregiune * j.id_regiune) as MAX, --24
    MIN(j.id_macroregiune * j.id_regiune) as MIN, --2
    SUM(j.id_macroregiune * j.id_regiune) as total --385
FROM
    judete j;

--3.    Sa se afiseze numarul de orase care au id_regiune mai mare de 2:

COUNT

SELECT COUNT(*) id_localitate  --9705
FROM localitati
WHERE id_regiune > 2; --


--4.    Sa se afiseze numarul de salarii (distincte) din tabela angajati.

SELECT COUNT (id_regiune) FROM localitati;--            13737
SELECT COUNT (DISTINCT id_regiune) FROM localitati;--   8


SELECT l.id_judet, J.DENUMIRE_JUDET, ROUND(AVG(l.id_localitate),0)
medie_localitati
FROM localitati l, judete J
```

```sql
WHERE L.ID_JUDET = J.ID_JUDET
GROUP BY L.id_judet, J.DENUMIRE_JUDET
ORDER BY medie_localitati;

/*
B   Bucuresti   1759
IL  Ialomita    5983
CV  Covasna 6194
SM  Satu Mare   6510
MH  Mehedinti   6510
IF  Ilfov   6518
TM  Timis   6521
DJ  Dolj    6591
CJ  Cluj    6615
AG  Arges   6618
BC  Bacau   6651
BT  Botosani    6733
SJ  Salaj   6758
MM  Maramures   6761
GJ  Gorj    6765
BH  Bihor   6828
HD  Hunedoara   6859
VS  Vaslui  6884
VL  Valcea  6888
SB  Sibiu   6898
IS  Iasi    6903
NT  Neamt   6905
BN  Bistrita-Nasaud 6912
SV  Suceava 6914
AB  Alba    6915
MS  Mures   6916
OT  Olt 6917
DB  Dambovita   6927
HR  Harghita    6977
CT  Constanta   7000
AR  Arad    7005
TL  Tulcea  7013
GR  Giurgiu 7059
BR  Braila  7073
BZ  Buzau   7076
GL  Galati  7141
TR  Teleorman   7184
CS  Caras-Severin   7215
BV  Brasov  7325
VN  Vrancea 7331
CL  Calarasi    7351
PH  Prahova 7363
*/

--8.    Sa se afiseze denumirea regiunilor si media regiunilor pentru
macroregiunile cu valoarea mai mare de 1.5

--HAVING

SELECT denumire_regiune, ROUND(AVG(id_regiune),1) as medie_regiune
FROM regiuni
GROUP BY denumire_regiune
HAVING ROUND(AVG(id_macroregiune),1)>1.5;

/*
Sud Vest - Oltenia  4
```

```
Sud Est 2
Vest    5
Bucuresti - Ilfov   8
Nord Est    1
Sud - Muntenia  3
*/


--10.   Sa se afiseze numai comenzile care au valoarea cuprinsa intre 1000
si 3000
--(conditia va fi mentionata in clauza HAVING deoarece se utilizeaza
functia de grup SUM):

SELECT
    r.denumire_regiune, ROUND(AVG(r.id_regiune),1) as medie_regiune,
    SUM(l.id_localitate * r.id_regiune) SUM --o suma oarecare, doar pentru
a exemplifica SUM
FROM
    regiuni r,
    localitati l
WHERE       r.id_regiune (+) = l.id_regiune
GROUP BY    r.denumire_regiune
HAVING      SUM(l.id_localitate * r.id_regiune) BETWEEN 1 AND 1000000000
ORDER BY    medie_regiune DESC;
```

--PARCURGEREA STRUCTURILOR IERARHICE

```
--1. Moduri de parcurgere a structurilor arborescente:
--
--•   TOP-DOWN – se construieste setul de inregistrari copil incepand cu
inregistrarea radacina
--•   BOTTOM-UP – se construieste setul de inregistrari parinte pana la
inregistrarea radacina pentru un anumit nivel din ierarhie
--•   DIRECT PE UN ANUMIT NIVEL – se construieste setul de inregistrari
incepand cu un anumit nivel din ierarhie
```

--I. Parcurgerea arborelui TOP-BOTTOM:

```
--1. Sa se afiseze localitatile si nivelul ierarhic al acestora pornind de
la localitate cu id-ul 568 (sa se ordoneze in functie de nivelul ierahic).
SELECT
    id_localitate,
    denumire_localitate,
    resedinta_judet,        --568   Balcani     1
    level
FROM
    localitati
CONNECT BY    PRIOR id_localitate = id_regiune
START WITH id_localitate = 568
ORDER BY LEVEL;



--Interogari ierarhice conditionate (clauza WHERE):
```

--Jonctiuni externe

--Functia DECODE si expresia CASE

```
--2.    Sa se indice in ce macroregiune se afla localitatile:
```

CASE

```
SELECT l.id_localitate, l.denumire_localitate,
CASE WHEN UPPER(m.id_macroregiune) = 1 THEN 'MACROREGIUNEA UNU'
WHEN UPPER(m.id_macroregiune) =  2 THEN 'MACROREGIUNEA DOI'
WHEN UPPER(m.id_macroregiune) = 3 THEN 'MACROREGIUNEA TREI'
WHEN UPPER(m.id_macroregiune) = 4 THEN 'MACROREGIUNEA PATRU'
ELSE '0' END macroregiuni --cifra stocata ca si caracter
FROM localitati l, macroregiuni m, regiuni r
where l.id_regiune = r.id_regiune
and R.id_macroregiune = M.id_macroregiune;

/*
615 Bals    MACROREGIUNEA PATRU
616 Balsa   MACROREGIUNEA PATRU
617 Balsoara    MACROREGIUNEA PATRU
618 Balta   MACROREGIUNEA PATRU
619 Balta   MACROREGIUNEA PATRU
620 Balta   MACROREGIUNEA PATRU
621 Balta Alba  MACROREGIUNEA DOI
622 Balta Arsa  MACROREGIUNEA DOI
623 Balta Doamnei   MACROREGIUNEA TREI
624 Balta Neagra    MACROREGIUNEA TREI
625 Balta Ratei MACROREGIUNEA DOI
626 Balta Sarata    MACROREGIUNEA TREI
627 Balta Tocila    MACROREGIUNEA DOI
*/
```

CASE

```
--Cu functia DECODE cerinta se poate rezolva astfel:
SELECT  l.id_localitate, l.denumire_localitate,
DECODE(UPPER(m.id_macroregiune) , 1 , 'MACROREGIUNEA UNU', 2 ,
'MACROREGIUNEA DOI', 3 , 'MACROREGIUNEA TREI', 4, 'MACROREGIUNEA PATRU', 0)
macroregiuni
FROM localitati l, macroregiuni m, regiuni r
where l.id_regiune = r.id_regiune
and R.id_macroregiune = M.id_macroregiune;

/*
568 Balcani MACROREGIUNEA DOI
569 Balcauti    MACROREGIUNEA DOI
570 Balcesti    MACROREGIUNEA UNU
571 Balcesti    MACROREGIUNEA UNU
572 Balcesti    MACROREGIUNEA PATRU
573 Balcesti    MACROREGIUNEA PATRU
574 Balciu  MACROREGIUNEA DOI
575 Balda   MACROREGIUNEA UNU
576 Baldana MACROREGIUNEA TREI
etc
*/
```

--Operatorii algebrei relationale UNION, INTERSECT, MINUS

```
/*
Sa se identifice cate localitati are fiecare judet, excluzand judetele care
incep cu litera M

operator MINUS
*/


SELECT
        j.denumire_judet,
        COUNT(l.denumire_localitate) numar_localitati,
        (
            CASE
                WHEN COUNT(l.denumire_localitate) BETWEEN 1  AND 25  THEN
'Judet_mic'
                WHEN COUNT(l.denumire_localitate) BETWEEN 26  AND 150
THEN 'Judet_mic'
                WHEN COUNT(l.denumire_localitate) > 150  THEN 'Judet_mare'
                ELSE 'judet_fara_localitati'
            END
        ) localitati
    FROM
        localitati l,
        judete j
    WHERE    l.id_judet = j.id_judet
    GROUP BY    j.denumire_judet
MINUS
SELECT
        j.denumire_judet,
        COUNT(l.denumire_localitate) numar_localitati,
        (
            CASE
                WHEN COUNT(l.denumire_localitate) BETWEEN 1  AND 25  THEN
'Judet_mic'
                WHEN COUNT(l.denumire_localitate) BETWEEN 26  AND 150
THEN 'Judet_mic'
                WHEN COUNT(l.denumire_localitate) > 150  THEN 'Judet_mare'
                ELSE 'judet_fara_localitati'
            END
        ) localitati
    FROM
        localitati l,
        judete j
    WHERE    l.id_judet = j.id_judet
            AND        j.denumire_judet LIKE 'M%'
    GROUP BY    j.denumire_judet
ORDER BY denumire_judet; --clauza Order by se poate mentiona o singura data
la sfarsitul intregii cereri.

/*
Bacau   507 Judet_mare
Bihor   458 Judet_mare
Bistrita-Nasaud 249 Judet_mare
Botosani    348 Judet_mare
Braila  144 Judet_mic
Brasov  165 Judet_mare
Bucuresti   1   Judet_mic
```

```
*/


--3.2.) Operatorul UNION – este utilizat pe 2 interogari pentru a reuni
inregistrarile selectate de prima interogare cu cele selectate de a doua
interogare (A + B).

/*
/*
Sa se identifice cate localitati sunt in fiecare macroregiune,
grupandu-le pe macroregiune si sortandu-le descendent dupa numarul de
localitati.

operator UNION
*/


SELECT m.denumire_macroregiune, COUNT(l.id_localitate) numar_localitati
    FROM localitati l, regiuni r, macroregiuni m
    WHERE l.id_regiune=r.id_regiune
    AND r.id_macroregiune=m.id_macroregiune
    GROUP BY m.denumire_macroregiune
    HAVING COUNT(l.id_localitate)BETWEEN 1  AND 25
UNION
SELECT m.denumire_macroregiune, COUNT(l.id_localitate) numar_localitati
    FROM localitati l, regiuni r, macroregiuni m
    WHERE l.id_regiune=r.id_regiune
    AND r.id_macroregiune=m.id_macroregiune
    GROUP BY m.denumire_macroregiune
    HAVING COUNT(l.id_localitate) BETWEEN 26  AND 150
UNION
SELECT m.denumire_macroregiune, COUNT(l.id_localitate) numar_localitati
    FROM localitati l, regiuni r, macroregiuni m
    WHERE l.id_regiune=r.id_regiune
    AND r.id_macroregiune=m.id_macroregiune
    GROUP BY m.denumire_macroregiune
    HAVING COUNT(l.id_localitate)>= 151
order by numar_localitati desc;

/*
MACROREGIUNEA DOI   4032
MACROREGIUNEA UNU   3891
MACROREGIUNEA PATRU 3585
MACROREGIUNEA TREI  2229
*/

--3.3.) Operatorul INTERSECT – este utilizat pe 2 interogari pentru a
returna
--doar inregistrarile comune selectate de prima interogare si cele
selectate de a doua interogare.

/*
Sa se evidentieze cate localitati au judetele a caror prima litera este M
si
sa se identifice ce fel de judet este (mic, mare, mediu.
*/
```

INTERSECT

```
SELECT
        j.denumire_judet,
        COUNT(l.denumire_localitate) numar_localitati,
        (
            CASE
                WHEN COUNT(l.denumire_localitate) BETWEEN 1  AND 25  THEN
'Judet_mic'
                WHEN COUNT(l.denumire_localitate) BETWEEN 26  AND 150
THEN 'Judet_mic'
                WHEN COUNT(l.denumire_localitate) > 150  THEN 'Judet_mare'
                ELSE 'judet_fara_localitati'
            END
        ) tip_judet
    FROM
        localitati l,
        judete j
    WHERE    l.id_judet = j.id_judet
    GROUP BY    j.denumire_judet
INTERSECT
SELECT
        j.denumire_judet,
        COUNT(l.denumire_localitate) numar_localitati,
        (
            CASE
                WHEN COUNT(l.denumire_localitate) BETWEEN 1  AND 25  THEN
'Judet_mic'
                WHEN COUNT(l.denumire_localitate) BETWEEN 26  AND 150
THEN 'Judet_mic'
                WHEN COUNT(l.denumire_localitate) > 150  THEN 'Judet_mare'
                ELSE 'judet_fara_localitati'
            END
        ) tip_judet
    FROM
        localitati l,
        judete j
    WHERE    l.id_judet = j.id_judet
            AND        j.denumire_judet LIKE 'M%'
    GROUP BY    j.denumire_judet
ORDER BY denumire_judet; --clauza Order by se poate mentiona o singura data
la sfarsitul intregii cereri.

/*
Maramures   247 Judet_mare
Mehedinti   358 Judet_mare
Mures   518 Judet_mare
*/
```

--GESTIUNEA ALTOR OBIECTE ALE BAZEI DE DATE

```
--
```

--TABELE VIRTUALE (VIEW)

```
--Tabele virtuale
--- Stocheaza interog?ri si permite reutilizarea  acestora
--- Protejeaza informa?iile de natura confidentiala
--- Protejeaza BD la actualizare
--- Tabele virtuale materializate stocheaza si inregistrarile
```

```
--Exemple:
--1.    Sa realizeze o tabela virtuala cu toate localitatile care incep cu
litera Z. 4. Actualizam id_regiune.

CREATE OR REPLACE VIEW v_localitati_Z
AS SELECT * FROM localitati
WHERE denumire_localitate like ('Z%');

SELECT * FROM v_localitati_Z;

UPDATE v_localitati_Z
SET id_regiune = id_regiune + 100;

rollback; --ca sa nu se salveze modificarile update-ului de mai sus.

--2.    Stocarea unei interogari care sa permita adaugarea unor conditii
ulterioare
SELECT * FROM v_localitati_Z
WHERE denumire_localitate like ('_a%')
AND resedinta_judet IS NOT NULL
;
--13612 Zalau   Y       6   SJ

--3.    Actualizarea tabelelor virtuale
CREATE OR REPLACE VIEW v_macroregiuni
AS SELECT id_macroregiune, denumire_macroregiune FROM macroregiuni;

UPDATE v_macroregiuni
SET id_macroregiune = id_macroregiune+100;

rollback;

--4.  Optiunea WITH READ ONLY
CREATE OR REPLACE VIEW v_macroregiuni
AS SELECT * FROM macroregiuni
WITH READ ONLY;

--5.    Sa se stearga tabela virtuala
DROP VIEW
DROP VIEW v_macroregiuni;

--6.    Vizualizarea informatiilor despre tabelele virtuale:
SELECT VIEW_NAME, TEXT FROM USER_VIEWS;

/*
--V_LOCALITATI_Z    "SELECT "ID_LOCALITATE","DENUMIRE_LOCALITATE",
"RESEDINTA_JUDET","RESEDINTA_REGIUNE","ID_REGIUNE","ID_JUDET"
FROM localitati
WHERE denumire_localitate like ('Z%')
*/

--INDECSI

-   Permit accesul rapid la date prin sortarea logica a înregistrarilor.
-   Se creaza automat la introducerea unei restrictii de cheie primara sau
de unicitate sau manual de catre utilizator.

--Exemple:
```

```
--1.    Sa se creeze un index pe tabela regiuni pe coloana
denumire_regiune:

SELECT * FROM regiuni WHERE denumire_regiune like ('N%');
--Cost 0.031
CREATE INDEX idx_denumire ON regiuni(denumire_regiune);
SELECT * FROM regiuni WHERE denumire_regiune like ('N%');
--Cost 0.028

--2.    Vizualizarea indecsilor unui anumit utilizator:

SELECT * FROM USER_INDEXES;

/*
NAME_ID_PK  NORMAL  RICHTERC_ID LISTA_INSTITUTII    TABLE   UNIQUE
DISABLED
LOC_REGIUNE_IX  NORMAL  RICHTERC_ID LOCALITATI  TABLE   NONUNIQUE
DISABLED
LOC_LOCALITATE_IX   NORMAL  RICHTERC_ID LOCALITATI  TABLE   NONUNIQUE
DISABLED
LOC_JUDET_IX    NORMAL  RICHTERC_ID LOCALITATI  TABLE   NONUNIQUE
DISABLED
LOC_ID_PK   NORMAL  RICHTERC_ID LOCALITATI  TABLE   UNIQUE  DISABLED
JUDET_C_ID_PK   NORMAL  RICHTERC_ID JUDETE  TABLE   UNIQUE  DISABLED
ID_REGIUNE_PK   NORMAL  RICHTERC_ID REGIUNI TABLE   UNIQUE  DISABLED
ID_MACROREGIUNE_PK  NORMAL  RICHTERC_ID MACROREGIUNI    TABLE   UNIQUE
DISABLED
IDX_DENUMIRE    NORMAL  RICHTERC_ID REGIUNI TABLE   NONUNIQUE   DISABLED
DEPT_ID_PK  NORMAL  RICHTERC_ID TIP_INSTITUTII  TABLE   UNIQUE  DISABLED
*/

--3.    Sa se stearga indexul creat anterior:
DROP INDEX
DROP INDEX idx_denumire;

--SECVENTE
--
--- Sunt utilizate pentru asigurarea unicitatii cheilor primare sau a
valorilor pentru care s-a impus o restrictie de tip UNIQUE.
--- Pot fi utilizate pentru mai multe tabele.
--- Pentru fiecare secventa se va preciza valoarea de inceput, pasul de
incrementare si valoarea maxima generata.

--Exemple:
--1.    Sa se creeze o secventa pentru asigurarea unicitatii cheii primare
din tabela Localitati.

CREATE SEQUENCE seq_idInstitutie
START WITH 500 INCREMENT BY 10
MAXVALUE 1000 NOCYCLE;

INSERT INTO lista_institutii VALUES (seq_idInstitutie.NEXTVAL,
'INEXIXTENT_TEST', '9999', NULL, '1');

--1 row inserted.

--2.    Sa se afiseze valoarea curenta a secventei:

SELECT seq_idInstitutie.CURRVAL FROM DUAL;
```

```
--3.    Sa se modifice pasul de incrementare si valoarea maxima pentru
secventa anterioara:
```

ALTER SEQUENCE

```
ALTER SEQUENCE seq_idInstitutie INCREMENT BY 100;
ALTER SEQUENCE seq_idInstitutie MAXVALUE 2000;

INSERT INTO lista_institutii VALUES (seq_idInstitutie.NEXTVAL,
'INEXIXTENT_TEST', '9999', NULL, '1');

--4.    Sa se vizualizeze informatiile depre secventele utilizatorilor:

SELECT * FROM USER_SEQUENCES;

--5.    Sa se stearga secventa seq_nrcomanda:
```

DROP SEQUENCE

```
DROP SEQUENCE seq_idInstitutie;

/*
```

SINONIME

```
-   Sunt nume alternative utilizate pentru referirea obiectelor unei baze
de date.
-   Pot fi sinonime publice (accesibile tuturor utilizatorilor) sau
private.
-   Sinonimele publice pot fi create numai de administratorul bazei de
date.

Exemple:

1.  Sa se creeze un sinonim pentru tabela lista_institutii:
*/
CREATE SYNONYM li FOR lista_institutii;

--2.    Vizualizarea sinonimelor se realizeaza astfel:

SELECT * FROM USER_SYNONYMS;

--3.    Sa se stearga sinonimul creat anterior:
```

DROP SYNONYM

```
DROP SYNONYM li;

--PARTITII
CREATE TABLE tabela_p(data DATE, cont VARCHAR2(50), divizia VARCHAR2(50))
PARTITION BY RANGE(data)
(PARTITION P1 VALUES LESS THAN (TO_DATE('01.04.2007','DD.MM.YYYY')),
PARTITION P2 VALUES LESS THAN (TO_DATE('01.09.2007','DD.MM.YYYY')));

INSERT INTO tabela_p SELECT data, cont, divizia FROM tabela_m;

SELECT * FROM tabela_p WHERE DATA <TO_DATE('01.02.2007','DD.MM.YYYY');
SELECT * FROM tabela_p  partition (p1) WHERE DATA
<TO_DATE('01.02.2007','DD.MM.YYYY');

--Cost 17
SELECT * FROM  tabela_m WHERE DATA <TO_DATE('01.02.2007','DD.MM.YYYY');
--Cost 99
```