

Proiect baze de date

Nedelcu Alexandru Daniel, grupa 1066

Descrierea bazei de date: obiectivul proiectului, descrierea tabelelor si a atributelor, precizarea restrictiilor si a tipurilor de legaturi (max. ½ pag.).

BD trebuie sa fie normalizata cel putin in FN3.

Baza de date creata de mine are ca scop stocarea informatiilor referitoare la politica macroeconomica a tarilor de pe planeta. Primele tabele create sunt:

CONTINENT, tabela ce contine un id, care este cheia primara, si o denumire ce are o restrictie de existenta.

TARA, tabelul cel mai important pentru baza mea, are ca attribute un nume cu o restrictie de existenta, capitala, anul de aderare la proiectul bazei de date, ca si cheia primara are un id tara.

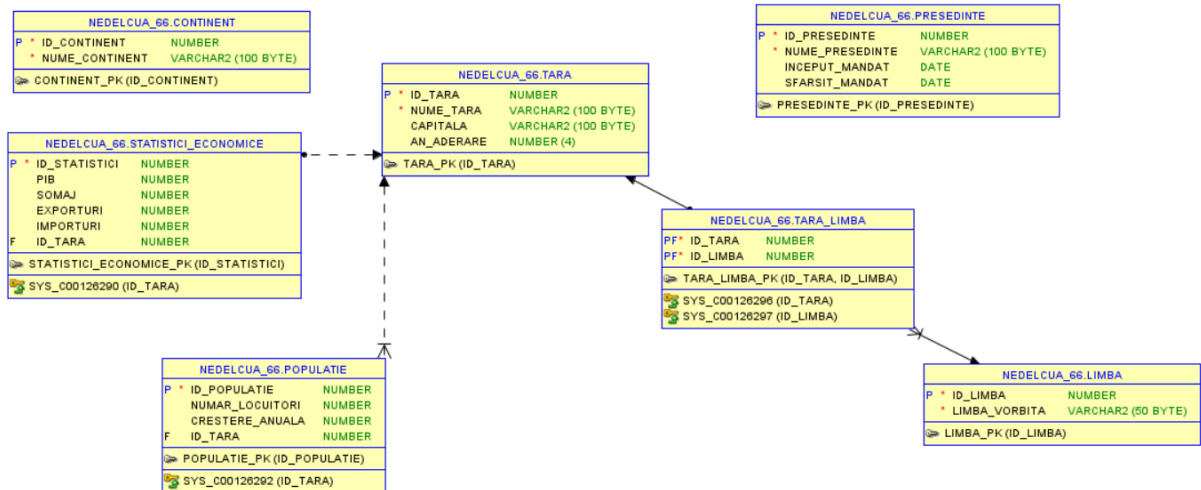
Tabelei **PRESEDINTE** ii sunt alocate 2 date, inceputul si sfarsitul mandatului, o cheia primara id presedinte, dar are si o restrictie la nume.

STATISTICI_ECONOMICE este tabela ce incapsuleaza domeniul economic al proiectului, are o cheia primara trecut ca id, multe attribute ce iau valori numerice, dar si o cheia externa ce face legatura cu tabela TARA. Asemnatoare este si tabela **POPULATIE**, care are un id cheia primara, o cheia externa catre TARA si cateva attribute numerice.

Tabela **LIMBA** este una simpla, ce contine o cheia primara si un atribut de tip text cu o restrictie de existenta. La final, mai exista tabela **TARA_LIMBA**, ce are o cheia primara proprie si 2 externe. Aceasta tabela a fost creata pentru a facilita relatia de many to many a tabelelor LIMBA si TARA. In schimb, intre CONTINENT si TARA exista o relatie de 1 la mai multi, adica un continent are mai multe tari, iar la restul tabelor in legatura cu TARA avem relatie 1 la 1.

Baza de date este normalizata in FN3 deoarece fiecare tabel are o cheia primară unică si nu există dependențe tranzitive.

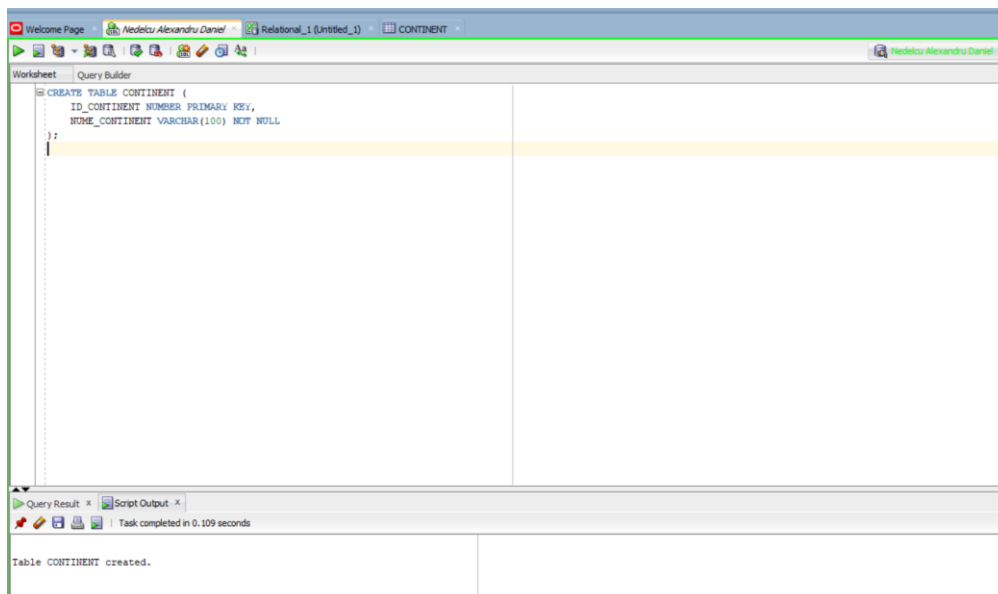
Schema corectă a bazei de date



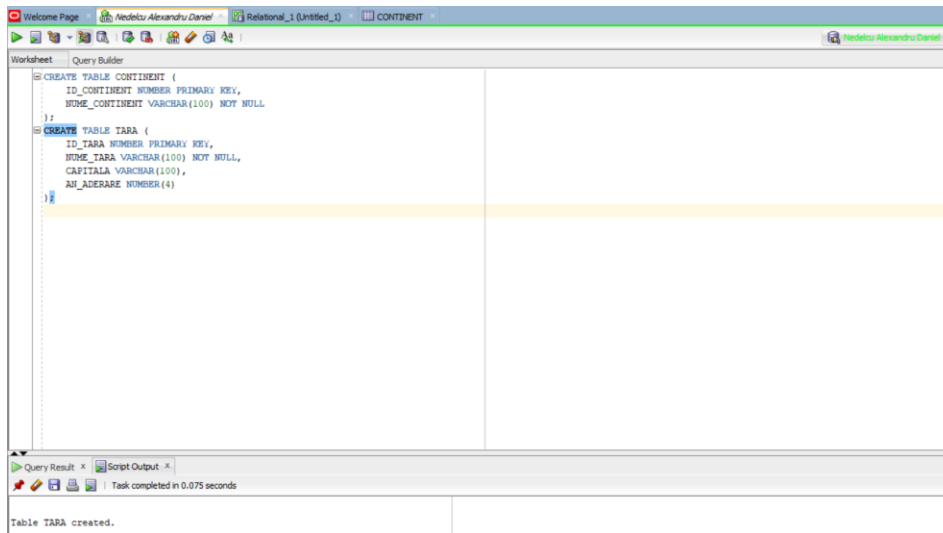
Crearea tabelelor

```

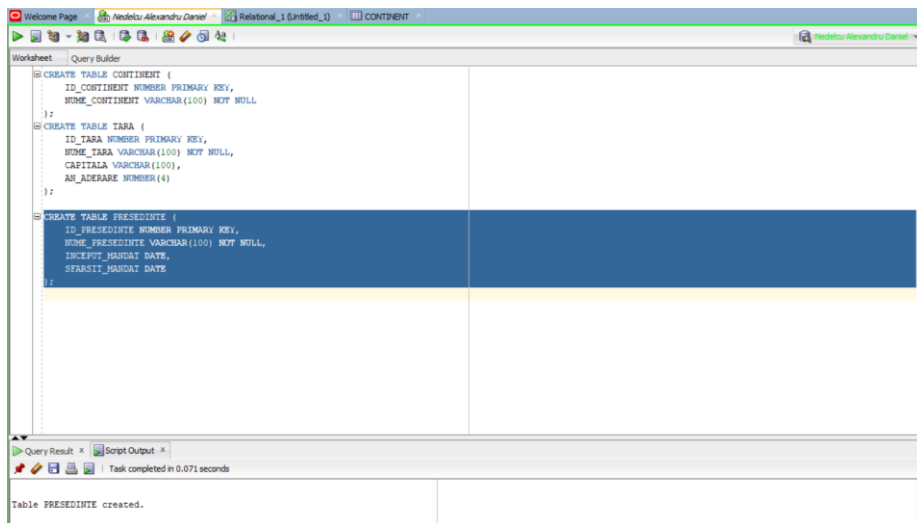
CREATE TABLE CONTINENT (
    ID_CONTINENT NUMBER PRIMARY KEY,
    NUME_CONTINENT VARCHAR(100) NOT NULL
);
    
```



```
CREATE TABLE TARA (  
    ID_TARA NUMBER PRIMARY KEY,  
    NUME_TARA VARCHAR(100) NOT NULL,  
    CAPITALA VARCHAR(100),  
    AN_ADERARE NUMBER(4)  
);
```



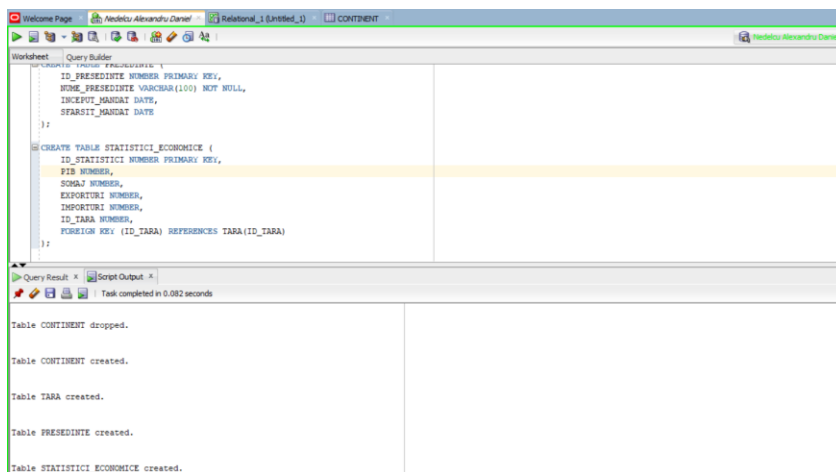
```
CREATE TABLE PRESEDINTE (  
    ID_PRESEDINTE NUMBER PRIMARY KEY,  
    NUME_PRESEDINTE VARCHAR(100) NOT NULL,  
    INCEPUT_MANDAT DATE,  
    SFARSIT_MANDAT DATE  
);
```



```

CREATE TABLE STATISTICI_ECONOMICE (
    ID_STATISTICI NUMBER PRIMARY KEY,
    PIB NUMBER,
    SOMAJ NUMBER,
    EXPORTURI NUMBER,
    IMPORTURI NUMBER,
    ID_TARA NUMBER,
    FOREIGN KEY (ID_TARA) REFERENCES TARA(ID_TARA)
);

```



```

CREATE TABLE POPULATIE (
    ID_POPULATIE NUMBER PRIMARY KEY,
    NUMAR_LOCUITORI NUMBER,
    CRESTERE_ANUALA NUMBER,

```

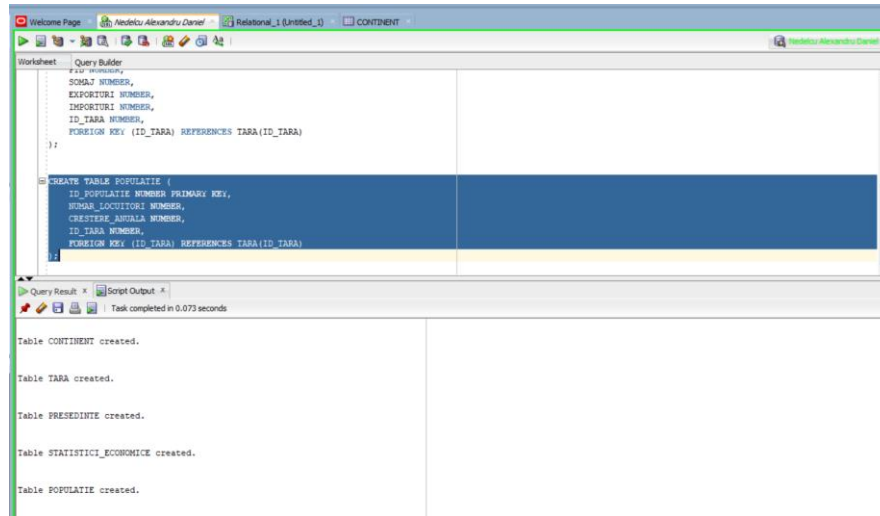
```

ID_TARA NUMBER,

FOREIGN KEY (ID_TARA) REFERENCES TARA(ID_TARA)

);

```



```

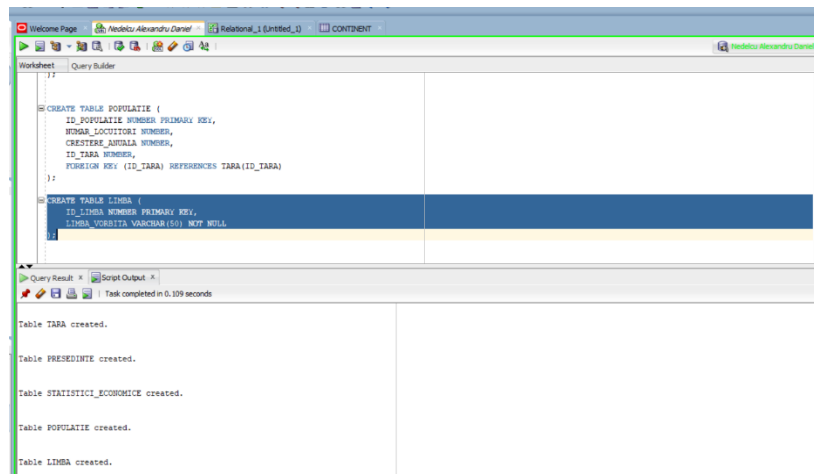
CREATE TABLE LIMBA (

    ID_LIMBA NUMBER PRIMARY KEY,

    LIMBA_VORBITA VARCHAR(50) NOT NULL

);

```



```

CREATE TABLE TARA_LIMBA (

    ID_TARA NUMBER,

    ID_LIMBA NUMBER,

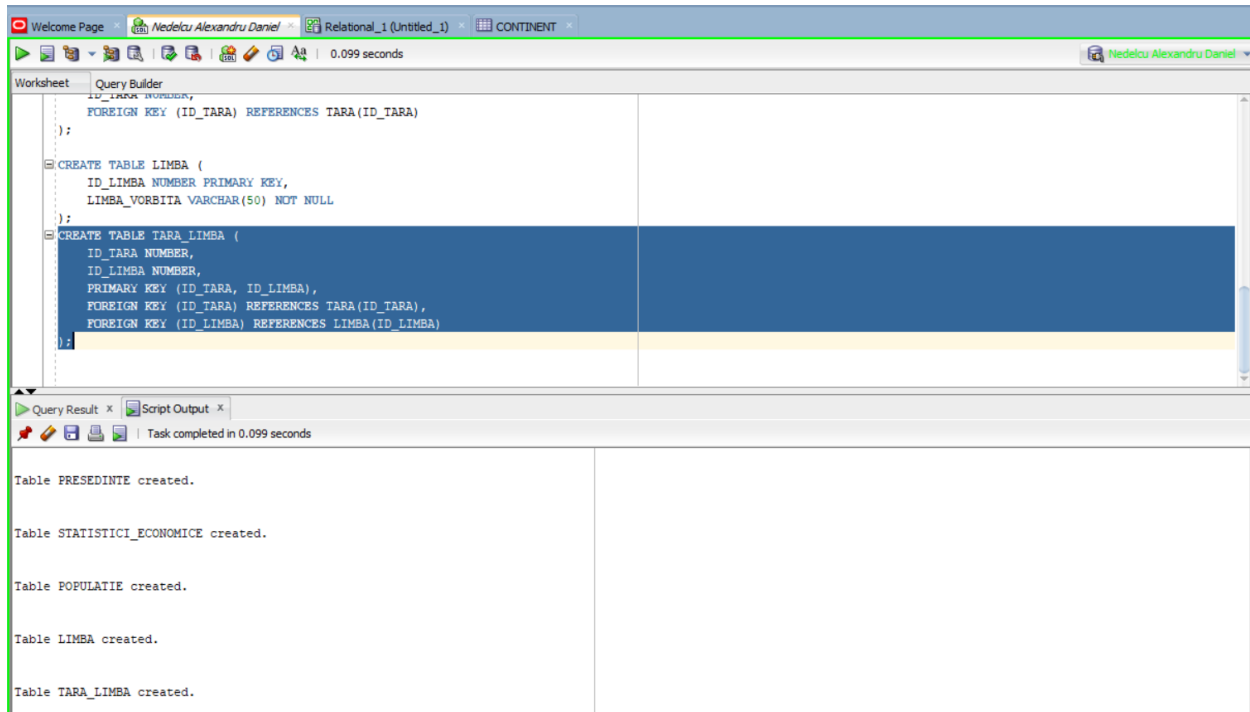
    PRIMARY KEY (ID_TARA, ID_LIMBA),

    FOREIGN KEY (ID_TARA) REFERENCES TARA(ID_TARA),

```

FOREIGN KEY (ID_LIMBA) REFERENCES LIMBA(ID_LIMBA)

);



Actualizarea structurii tabelelor si modificarea restrictiilor de integritate

Sa se adauge atributul datoriei in tabela STATISTICI_ECONOMICE.

ALTER TABLE STATISTICI_ECONOMICE

ADD DATORII NUMBER;

Sa se adauge in tabela TARA o cheie externa la tabela CONTINENT.

ALTER TABLE TARA

ADD ID_CONTINENT NUMBER;

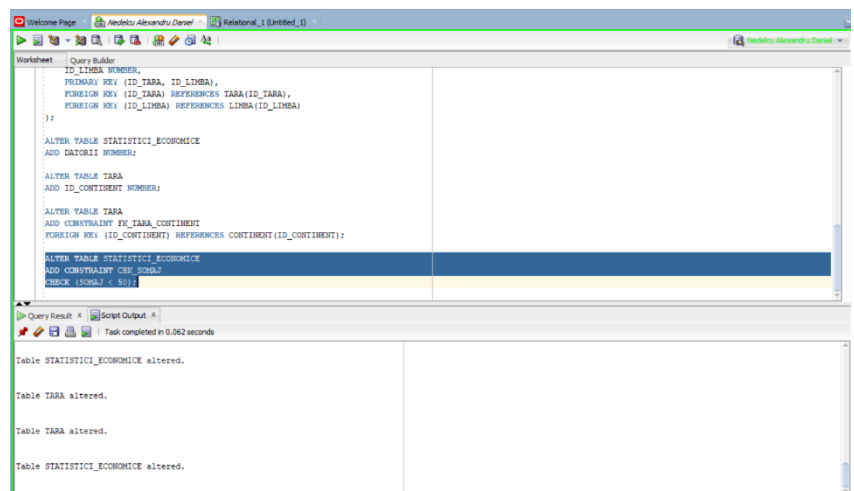
ALTER TABLE TARA

ADD CONSTRAINT FK_TARA_CONTINENT

FOREIGN KEY (ID_CONTINENT)
REFERENCES
CONTINENT(ID_CONTINENT);

Sa se adauge o restrictie de integritate de tip check asupra somajului.

ALTER TABLE STATISTICI_ECONOMICE



ADD CONSTRAINT CHK_SOMAJ

CHECK (SOMAJ < 50);

Sa se adauge o restrictie de tipul not null pentru campul capitala din TARA.

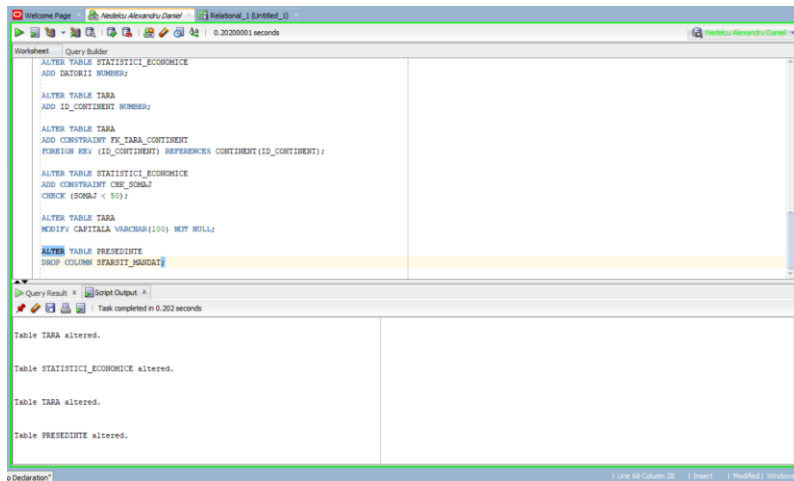
ALTER TABLE TARA

MODIFY CAPITALA VARCHAR(100) NOT NULL;

Sa se stearga campul sfarsit_mandat din tabela PRESEDINTE.

ALTER TABLE PRESEDINTE

DROP COLUMN SFARSIT_MANDAT;



Redenumiti tabela STATISTICI_ECONOMICE.

RENAME STATISTICI_ECONOMICE TO STATISTICI;

Sa se adauge coloanele rural si urban, de tip number(3) in tabela POPULATIE, apoi sa li se puna restrictia rural+urban=100.

ALTER TABLE POPULATIE

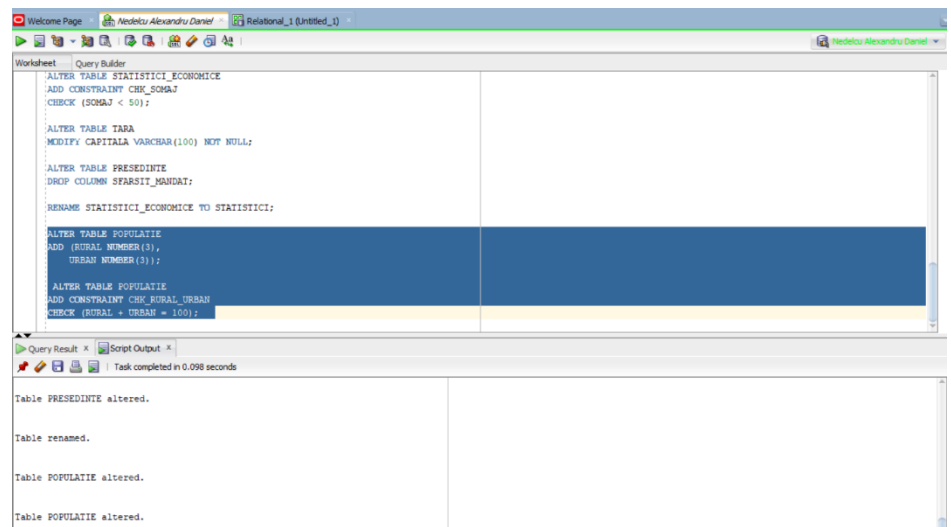
ADD (RURAL NUMBER(3),

URBAN NUMBER(3));

ALTER TABLE POPULATIE

ADD CONSTRAINT
CHK_RURAL_URBAN

CHECK (RURAL + URBAN =
100);



Adăugarea (min 10, max 20) de înregistrări în fiecare tabelă

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (1, 'Europa');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (2, 'Asia');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (4, 'America de nord');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (3, 'America de sud');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (5, 'Africa');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (6, 'Oceania');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (7, 'Antartica');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

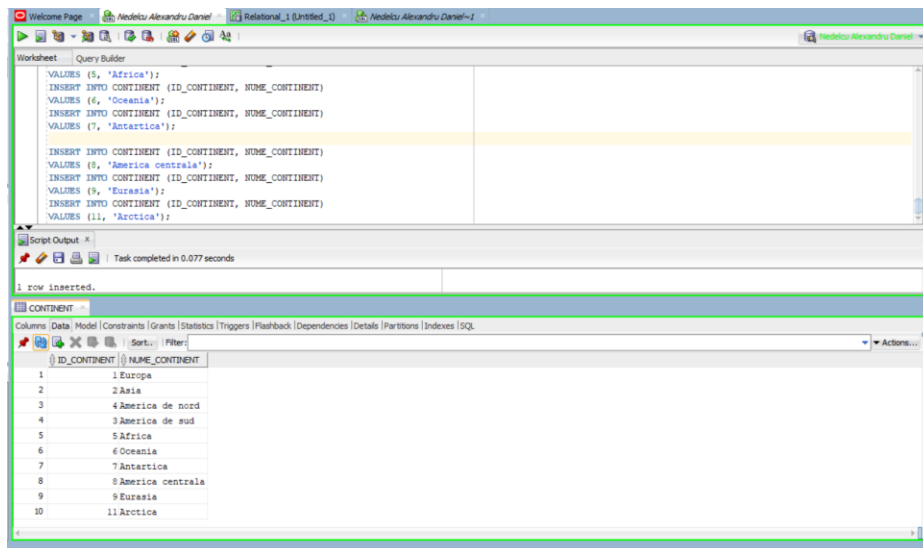
VALUES (8, 'America centrala');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (9, 'Eurasia');

INSERT INTO CONTINENT (ID_CONTINENT, NUME_CONTINENT)

VALUES (11, 'Arctica');




```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (10, 'Chineza');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (101, 'Japoneza');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (5, 'Engleza');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (15, 'Romana');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (62, 'Franceza');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (2, 'Portugheza');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (22, 'Spaniola');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
VALUES (1500, 'Araba');
```

```
INSERT INTO LIMBA (ID_LIMBA,
LIMBA_VORBITA)
```

```
VALUES (33, 'Estona');
```

```
INSERT INTO LIMBA (ID_LIMBA,
LIMBA_VORBITA)
```

```
VALUES (31, 'Daneza');
```

```
INSERT INTO LIMBA (ID_LIMBA,
LIMBA_VORBITA)
```

```
VALUES (32, 'Suedeza');
```

```
INSERT INTO LIMBA (ID_LIMBA,
LIMBA_VORBITA)
```

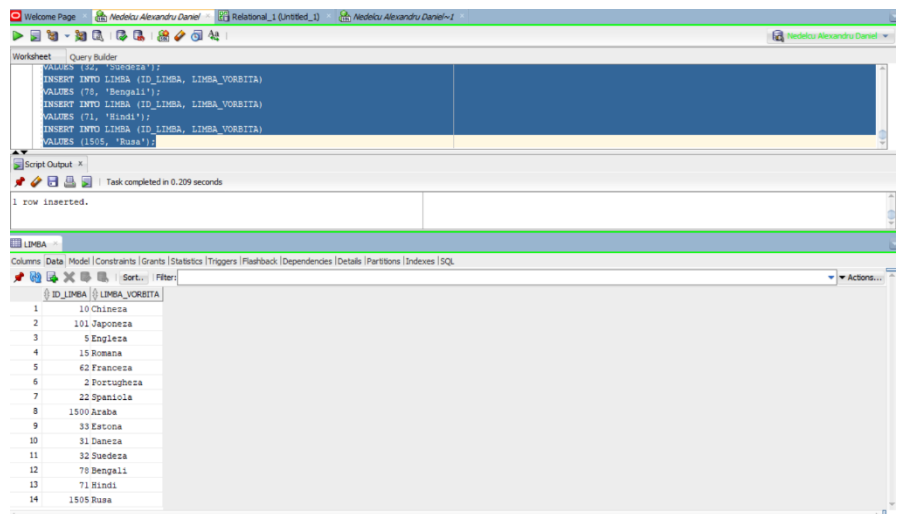
```
VALUES (78, 'Bengali');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
```

```
VALUES (71, 'Hindi');
```

```
INSERT INTO LIMBA (ID_LIMBA, LIMBA_VORBITA)
```

```
VALUES (1505, 'Rusa');
```



The screenshot shows a database management tool interface. The top pane displays a SQL script with multiple INSERT statements for the LIMBA table. The bottom pane shows the results of the script, indicating that 1 row was inserted. Below the results, a table view of the LIMBA table is shown, listing 14 rows of data.

ID_LIMBA	LIMBA_VORBITA
1	10 Chineza
2	101 Japoneza
3	5 Engleza
4	15 Romana
5	62 Franceza
6	2 Portugheza
7	22 Spaniola
8	1500 Araba
9	33 Estona
10	31 Daneza
11	32 Suedeza
12	78 Bengali
13	71 Hindi
14	1505 Rusa

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (1, 'Romania', 'Bucuresti', 2005,1);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (2, 'Danemarca', 'Copenhaga', 2005,1);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (3, 'Spania', 'Madrid', 2008,1);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (4, 'Franta', 'Paris', 2006,1);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (5, 'SUA', 'Washington DC', 2021,4);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (6, 'India', 'New Delhi', 2017,2);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (7, 'Australia', 'Canberra', 2009,6);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (8, 'Argentina', 'Buenos Aires', 2005,3);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (9, 'Yemen', 'Sanaa', 2014,2);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (10, 'Rusia', 'Moscova', 2023,9);
```

```
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (11, 'Egipt', 'Cairo', 2010,5);
```

The screenshot shows a database management tool interface. The top section displays a SQL script with 11 INSERT statements for a table named TARA. The script is as follows:

```
VALUES (1, 'Romania', 'Bucuresti', 2005,1);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (2, 'Danemarca', 'Copenhaga', 2005,1);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (3, 'Spania', 'Madrid', 2008,1);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (4, 'Franta', 'Paris', 2006,1);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (5, 'SUA', 'Washington DC', 2021,4);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (6, 'India', 'New Delhi', 2017,2);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (7, 'Australia', 'Canberra', 2009,6);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (8, 'Argentina', 'Buenos Aires', 2005,3);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (9, 'Yemen', 'Sanaa', 2014,2);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (10, 'Rusia', 'Moscova', 2023,9);
INSERT INTO TARA (ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, ID_CONTINENT)
VALUES (11, 'Egipt', 'Cairo', 2010,5);
```

Below the script, the 'Script Output' section shows the execution results:

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

The bottom section displays the 'TARA' table structure and its contents:

ID_TARA	NUME_TARA	CAPITALA	AN_ADERARE	ID_CONTINENT
1	Romania	Bucuresti	2005	1
2	Danemarca	Copenhaga	2005	1
3	Spania	Madrid	2008	1
4	Franta	Paris	2006	1
5	SUA	Washington DC	2021	4
6	India	New Delhi	2017	2
7	Australia	Canberra	2009	6
8	Argentina	Buenos Aires	2005	3
9	Yemen	Sanaa	2014	2
10	Rusia	Moscova	2023	9
11	Egipt	Cairo	2010	5

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (1, 'Mircea Daniel');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (2, 'Yoshie Ootobong');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (3, 'Harriet Fry');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (4, 'Rene Moore');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (5, 'Orville Wolfe');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (6, 'Colton Wilson');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (7, 'Vera Greer');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (8, 'Rosario Clay');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (9, 'Rowena Hamilton');
```

```
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (10, 'Lacy Parks');
```

The screenshot displays a database management interface. The top section shows a SQL query window with the following content:

```
VALUES (6, 'Colton Wilson');
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (7, 'Vera Greer');
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (8, 'Rosario Clay');
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (9, 'Rowena Hamilton');
INSERT INTO PRESEDINTE (ID_PRESEDINTE, NUME_PRESEDINTE)
VALUES (10, 'Lacy Parks');
```

Below the query window, the 'Script Output' tab shows the execution results:

```
Task completed in 0.148 seconds
1 row inserted.
1 row inserted.
```

The bottom section shows a table view for the 'PRESEDINTE' table. The table has three columns: 'ID_PRESEDINTE', 'NUME_PRESEDINTE', and 'INCEPUT_MANDAT'. The data is as follows:

ID_PRESEDINTE	NUME_PRESEDINTE	INCEPUT_MANDAT
1	Mircea Daniel	(null)
2	Yoshie Ootobong	(null)
3	Harriet Fry	(null)
4	Rene Moore	(null)
5	Orville Wolfe	(null)
6	Colton Wilson	(null)
7	Vera Greer	(null)
8	Rosario Clay	(null)
9	Rowena Hamilton	(null)
10	Lacy Parks	(null)

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (1, 19000000, 50000, 1, 62, 38);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (2, 149000000, 420000, 2, 50, 50);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (3, 2000000, 57000, 3, 70, 30);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (4, 9800000, 907000, 4, 71, 29);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (5, 20000, 5000, 5, 70, 30);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (6, 52350000, 53000, 6, 2, 90);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (7, 640000, 5000, 7, 50, 50);
```

```
INSERT INTO POPULATIE (ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA,  
RURAL, URBAN)
```

```
VALUES (8, 6600000, 17000, 8, 70, 30);
```

```
INSERT INTO POPULATIE  
(ID_POPULATIE,  
NUMAR_LOCUITORI,  
CRESTERE_ANUALA,  
ID_TARA, RURAL, URBAN)
```

```
VALUES (9, 7420000, 57000, 9,  
60, 40);
```

```
INSERT INTO POPULATIE  
(ID_POPULATIE,  
NUMAR_LOCUITORI,  
CRESTERE_ANUALA,  
ID_TARA, RURAL, URBAN)
```

```
VALUES (10, 4000000, 57000,  
10, 70, 30);
```

The screenshot shows a database management tool interface. The top section displays a list of SQL queries in a 'Query Builder' window. Below this, a 'Script Output' window shows the execution status: 'Task completed in 0.177 seconds'. The bottom section displays a table named 'POPULATIE' with columns: ID_POPULATIE, NUMAR_LOCUITORI, CRESTERE_ANUALA, ID_TARA, RURAL, and URBAN. The table contains 10 rows of data.

ID_POPULATIE	NUMAR_LOCUITORI	CRESTERE_ANUALA	ID_TARA	RURAL	URBAN
1	19000000	50000	1	62	38
2	149000000	420000	2	50	50
3	2000000	57000	3	70	30
4	9800000	907000	4	71	29
5	20000	5000	5	70	30
6	52350000	53000	6	2	90
7	640000	5000	7	50	50
8	6600000	17000	8	70	30
9	7420000	57000	9	60	40
10	4000000	57000	10	70	30

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (10, 7422442, 11, 421, 321, 1, 0);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (11, 982442, 25, 2421, 3231, 2, 1110);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (12, 97696876, 43, 151, 26, 3, 5125);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (13, 832442, 21, 9087, 5321, 4, 42141);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (15, 22442, 11, 21, 31, 5, 0);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (16, 255232442, 2, 51421, 52321, 6, 97543);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (17, 4222442, 6, 6511, 8321, 7, 0);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (18, 22442, 1, 21, 21, 8, 5);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (19, 2442, 37, 5151, 6321, 9, 4215);

INSERT INTO STATISTICI (ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, DATORII)
VALUES (50, 6161244242, 15, 452121, 366421, 10, 51515);

The screenshot shows a database management tool interface. The top section displays a list of SQL queries in a blue box, including INSERT and VALUES statements for the STATISTICI table. Below the queries, a 'Script Output' section shows the message '1 row inserted.' and 'Task completed in 0.143 seconds'. The bottom section shows a table view of the STATISTICI table with columns ID_STATISTICI, PIB, SOMAJ, EXPORTURI, IMPORTURI, ID_TARA, and DATORII. The table contains 10 rows of data, corresponding to the queries above.

ID_STATISTICI	PIB	SOMAJ	EXPORTURI	IMPORTURI	ID_TARA	DATORII
10	7422442	11	421	321	1	0
11	982442	25	2421	3231	2	1110
12	97696876	43	151	26	3	5125
13	832442	21	9087	5321	4	42141
15	22442	11	21	31	5	0
16	255232442	2	51421	52321	6	97543
17	4222442	6	6511	8321	7	0
18	22442	1	21	21	8	5
19	2442	37	5151	6321	9	4215
50	6161244242	15	452121	366421	10	51515

Actualizarea inregistrarilor

Sa se stearga din tabela continente acelea care nu apar niciodata in tabela TARA.

```
DELETE FROM CONTINENT
```

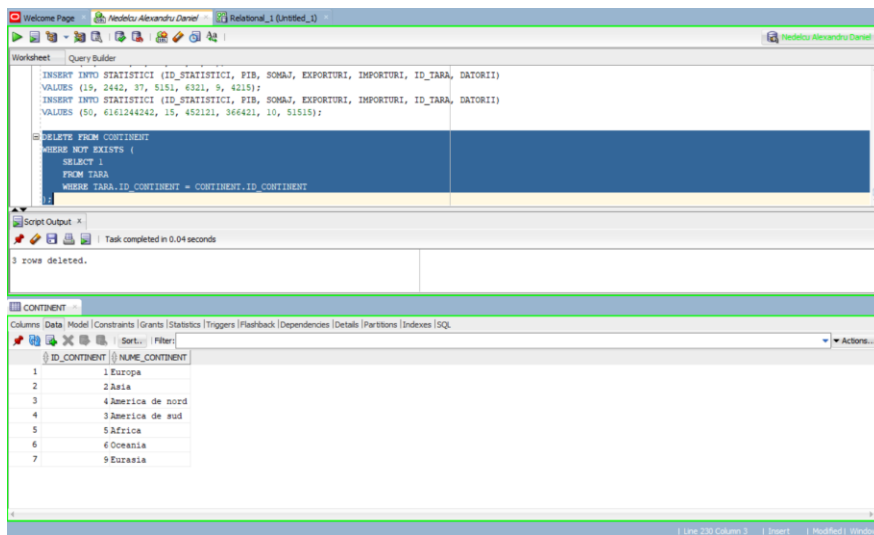
```
WHERE NOT EXISTS (
```

```
SELECT 1
```

```
FROM TARA
```

```
WHERE TARA.ID_CONTINENT = CONTINENT.ID_CONTINENT
```

```
);
```



Adauga coloana EXPORT-IMPORT ce este number in tabela STATISTICI, initializeaza cu 0 apoi calculeaza.

```
ALTER TABLE STATISTICI
```

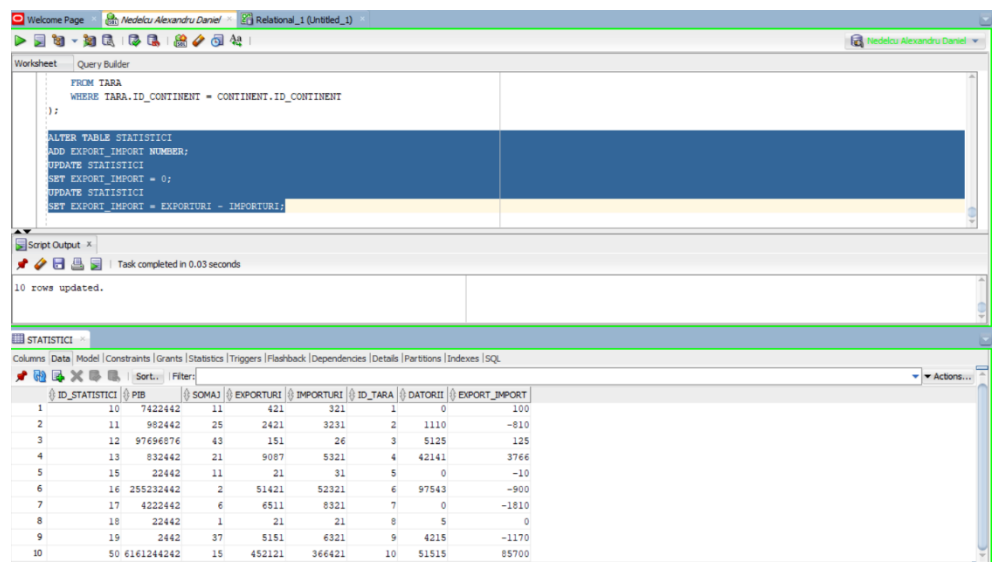
```
ADD EXPORT_IMPORT NUMBER;
```

```
UPDATE STATISTICI
```

```
SET EXPORT_IMPORT = 0;
```

```
UPDATE STATISTICI
```

```
SET EXPORT_IMPORT =  
EXPORTURI -  
IMPORTURI;
```



Stergerea si recuperarea unei tabele

DROP TABLE PRESEDINTE;

The screenshot shows the SQL Developer interface. The 'Script Output' pane displays the message '10 rows updated.' Below the script editor, the 'Messages - Log' pane shows the error 'ORA-00942: table or view does not exist'. The 'PRESEDINTE' table structure is visible in the 'Columns' pane, showing columns: ID_PRESEDINTE, NUME_PRESEDINTE, and INCEPUT_MANDAT.

```
FROM TARA
WHERE TARA.ID_CONTINENT = CONTINENT.ID_CONTINENT
);

ALTER TABLE STATISTICI
ADD EXPORT_IMPORT NUMBER;
UPDATE STATISTICI
SET EXPORT_IMPORT = 0;
UPDATE STATISTICI
SET EXPORT_IMPORT = EXPORTURI - IMPORTURI;

DROP TABLE PRESEDINTE;
```

Script Output x
Task completed in 0.115 seconds

10 rows updated.

PRESEDINTE

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

ID_PRESEDINTE NUME_PRESEDINTE INCEPUT_MANDAT

Messages - Log

ORA-00942: table or view does not exist

Messages | Logging Page | Statements

FLASHBACK TABLE PRESEDINTE TO BEFORE DROP;

The screenshot shows the SQL Developer interface. The 'Script Output' pane displays the message 'Table PRESEDINTE dropped.' and 'Flashback succeeded.' Below the script editor, the 'PRESEDINTE' table structure is visible in the 'Columns' pane, showing columns: ID_PRESEDINTE, NUME_PRESEDINTE, and INCEPUT_MANDAT. The table data is displayed in the 'Data' pane, showing 5 rows.

```
);

ALTER TABLE STATISTICI
ADD EXPORT_IMPORT NUMBER;
UPDATE STATISTICI
SET EXPORT_IMPORT = 0;
UPDATE STATISTICI
SET EXPORT_IMPORT = EXPORTURI - IMPORTURI;

DROP TABLE PRESEDINTE;

FLASHBACK TABLE PRESEDINTE TO BEFORE DROP;
```

Script Output x
Task completed in 0.109 seconds

Table PRESEDINTE dropped.

Flashback succeeded.

PRESEDINTE

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

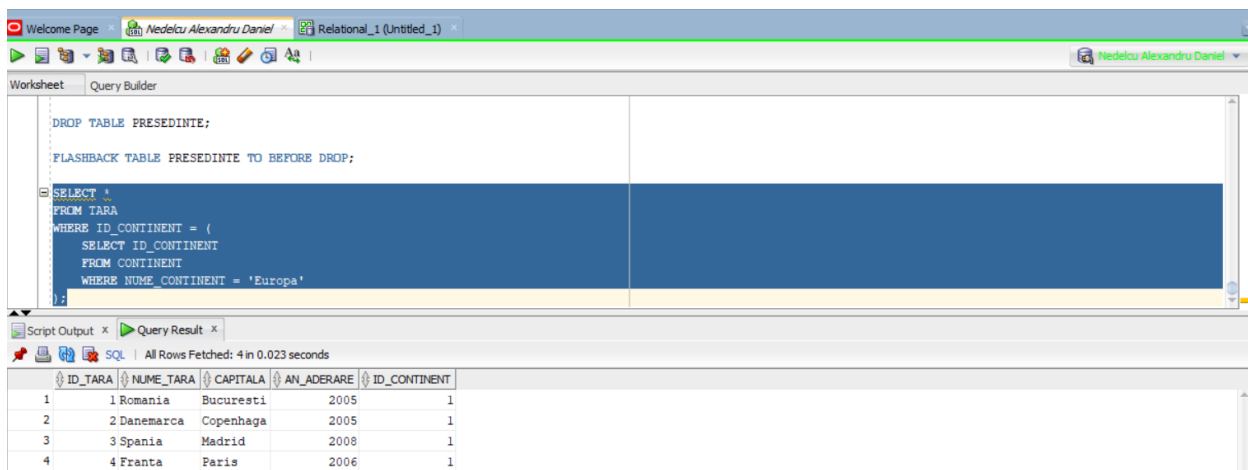
ID_PRESEDINTE NUME_PRESEDINTE INCEPUT_MANDAT

	ID_PRESEDINTE	NUME_PRESEDINTE	INCEPUT_MANDAT
1	1	Mircea Daniel	(null)
2	2	Yoshie Otobong	(null)
3	3	Harriet Fry	(null)
4	4	Rene Moore	(null)
5	5	Orville Wolfe	(null)

Exemple de interogări variate (min 20) – incluzând și operatorii UNION, INTERSECT, MINUS, expresiile DECODE și CASE, cereri imbricate, diverse funcții single-row, funcții de grup, structuri ierarhice, jonctiuni

Sa se selecteze numai tarile din Europa.

```
SELECT *  
FROM TARA  
WHERE ID_CONTINENT = (  
    SELECT ID_CONTINENT  
    FROM CONTINENT  
    WHERE NUME_CONTINENT = 'Europa'  
);
```



The screenshot shows a database query tool interface. The top bar includes a 'Welcome Page' button and a user profile 'Nedelcu Alexandru Daniel'. Below the toolbar, the 'Query Builder' tab is active, displaying the following SQL query:

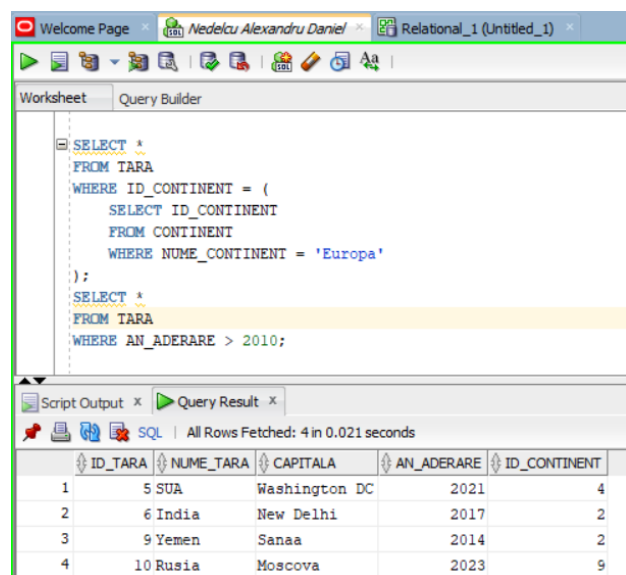
```
DROP TABLE PRESEDINTE;  
FLASHBACK TABLE PRESEDINTE TO BEFORE DROP;  
  
SELECT *  
FROM TARA  
WHERE ID_CONTINENT = (  
    SELECT ID_CONTINENT  
    FROM CONTINENT  
    WHERE NUME_CONTINENT = 'Europa'  
);
```

The 'Script Output' tab shows the query results, indicating 'All Rows Fetched: 4 in 0.023 seconds'. The results are displayed in a table with the following columns: ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, and ID_CONTINENT.

ID_TARA	NUME_TARA	CAPITALA	AN_ADERARE	ID_CONTINENT
1	1 Romania	Bucuresti	2005	1
2	2 Danemarca	Copenhaga	2005	1
3	3 Spania	Madrid	2008	1
4	4 Franta	Paris	2006	1

Sa se selecteze toate tarile care au anul de aderare dupa 2010.

```
SELECT *  
FROM TARA  
WHERE AN_ADERARE > 2010;
```



The screenshot shows the same database query tool interface. The 'Query Builder' tab displays the following SQL query:

```
SELECT *  
FROM TARA  
WHERE ID_CONTINENT = (  
    SELECT ID_CONTINENT  
    FROM CONTINENT  
    WHERE NUME_CONTINENT = 'Europa'  
);  
  
SELECT *  
FROM TARA  
WHERE AN_ADERARE > 2010;
```

The 'Script Output' tab shows the query results, indicating 'All Rows Fetched: 4 in 0.021 seconds'. The results are displayed in a table with the following columns: ID_TARA, NUME_TARA, CAPITALA, AN_ADERARE, and ID_CONTINENT.

ID_TARA	NUME_TARA	CAPITALA	AN_ADERARE	ID_CONTINENT
1	5 SUA	Washington DC	2021	4
2	6 India	New Delhi	2017	2
3	9 Yemen	Sanaa	2014	2
4	10 Rusia	Moscova	2023	9

Afiseaza id-ul tarilor care au somajul mai mic de 5 si mai mare de 30.

```
SELECT ID_TARA
FROM STATISTICI
WHERE SOMAJ < 5
UNION
SELECT ID_TARA
FROM STATISTICI
WHERE SOMAJ > 30;
```

The screenshot shows a SQL query editor with the following query:

```
);
SELECT *
FROM TARA
WHERE AN_ADERARE > 2010;
SELECT ID_TARA
FROM STATISTICI
WHERE SOMAJ < 5
UNION
SELECT ID_TARA
FROM STATISTICI
WHERE SOMAJ > 30;
```

The results pane shows the following data:

ID_TARA	
1	6
2	8
3	3
4	9

Afiseaza cu ajutorul operatorului intersect id-ul tarilor care au pib-ul mai mic decat 100000 si cresterea anuala mai mica decat 99999.

```
SELECT ID_TARA
FROM STATISTICI
WHERE PIB < 100000
INTERSECT
SELECT ID_TARA
FROM POPULATIE
WHERE CRESTERE_ANUALA < 99999;
```

The screenshot shows a SQL query editor with the following query:

```
UNION
SELECT ID_TARA
FROM STATISTICI
WHERE SOMAJ > 30;
SELECT ID_TARA
FROM STATISTICI
WHERE PIB < 100000
INTERSECT
SELECT ID_TARA
FROM POPULATIE
WHERE CRESTERE_ANUALA < 99999;
```

The results pane shows the following data:

ID_TARA	
1	5
2	8
3	9

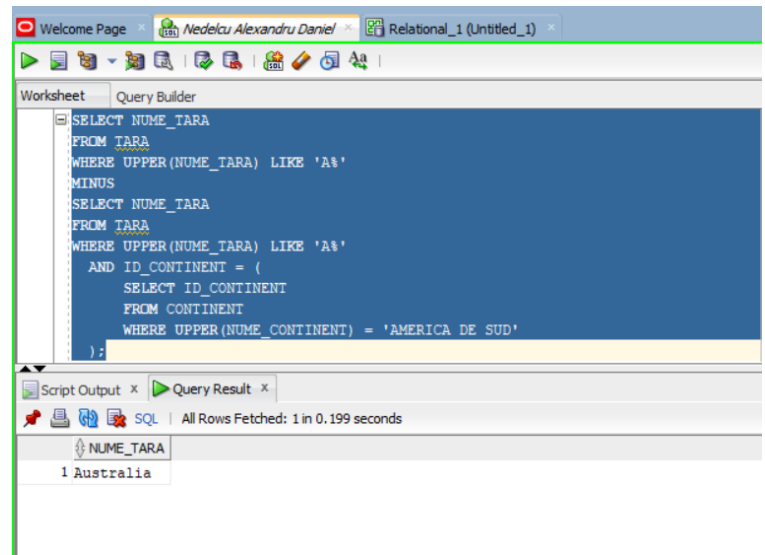
Selecteaza toate tarile care incep cu A fara pe cele din America de Sud folosindu-te de MINUS.

```
SELECT NUME_TARA
FROM TARA
WHERE UPPER(NUME_TARA) LIKE 'A%'
MINUS
SELECT NUME_TARA
```

```

FROM TARA
WHERE UPPER(NUME_TARA) LIKE 'A%'
AND ID_CONTINENT = (
    SELECT ID_CONTINENT
    FROM CONTINENT
    WHERE UPPER(NUME_CONTINENT) =
'AMERICA DE SUD'
);

```



Creaza o noua coloana numita deficit in tabela Statistici. Folosindu-te de case sau decode, initiaza deficitul astfel, daca datorii=0 deficit=0, daca datorii <=3000 atunci deficit=1 si deficit=-1 altfel.

```

ALTER TABLE STATISTICI
ADD DEFICIT NUMBER;
UPDATE STATISTICI
SET DEFICIT =
CASE
    WHEN DATORII = 0 THEN 0
    WHEN DATORII <= 3000 THEN 1
    ELSE -1
END;

```

The screenshot shows a SQL query in a 'Query Builder' window. The query is:

ALTER TABLE STATISTICI

ADD DEFICIT NUMBER;

UPDATE STATISTICI

SET DEFICIT =

CASE

 WHEN DATORII = 0 THEN 0

 WHEN DATORII <= 3000 THEN 1

 ELSE -1

END;

The 'Query Result' tab shows 'Table STATISTICI altered.'

Below the query editor, a table view of the 'STATISTICI' table is shown. The table has 10 columns: ID_STATISTICI, PIB, SOMA, EXPORTURI, IMPORTURI, ID_TARA, DATORII, EXPORT_IMPORT, and DEFICIT. The data is as follows:

ID_STATISTICI	PIB	SOMA	EXPORTURI	IMPORTURI	ID_TARA	DATORII	EXPORT_IMPORT	DEFICIT
1	10	7422442	11	421	321	1	0	100
2	11	982442	25	2421	3231	2	1110	-810
3	12	97696876	43	151	26	3	5125	125
4	13	832442	21	9087	5321	4	42141	3766
5	15	22442	11	21	31	5	0	-10
6	16	255232442	2	51421	52321	6	97543	-900
7	17	4222442	6	6511	8321	7	0	-1810
8	18	22442	1	21	21	8	5	0
9	19	2442	37	5151	6321	9	4215	-1170
10	50	6161244242	15	452121	366421	10	51515	85700

Modifica coloana Export_IMPORT ca fiind de tipul varchar2. Dupa ce se transforma in variabila nula, se actualizeaza astfel, daca $\text{Export_Import} < 0$ coloana in Export_IMPORT vei scrie negativ, altfel pozitiv.

```
UPDATE STATISTICI
```

```
SET EXPORT_IMPORT = NULL;
```

```
ALTER TABLE STATISTICI
```

```
MODIFY EXPORT_IMPORT VARCHAR2(10);
```

```
SELECT EXPORTURI, IMPORTURI,
```

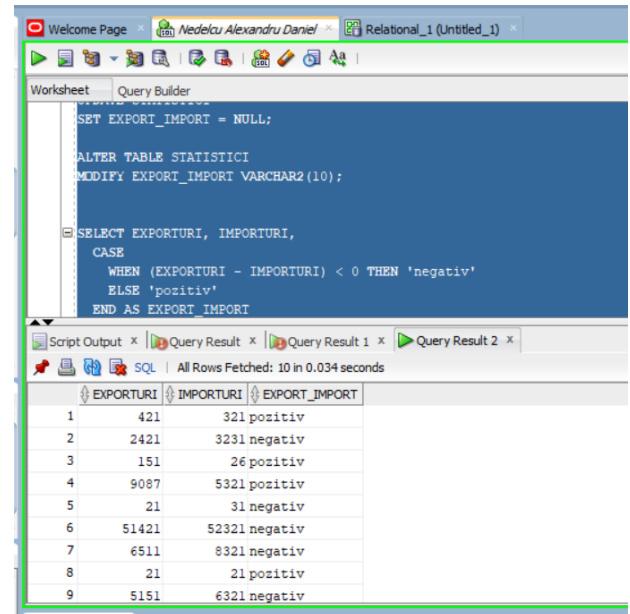
```
CASE
```

```
  WHEN (EXPORTURI - IMPORTURI) < 0 THEN 'negativ'
```

```
  ELSE 'pozitiv'
```

```
END AS EXPORT_IMPORT
```

```
FROM STATISTICI;
```



The screenshot shows the SQL Developer interface with a query window containing the following SQL code:

```
SET EXPORT_IMPORT = NULL;

ALTER TABLE STATISTICI
MODIFY EXPORT_IMPORT VARCHAR2(10);

SELECT EXPORTURI, IMPORTURI,
CASE
  WHEN (EXPORTURI - IMPORTURI) < 0 THEN 'negativ'
  ELSE 'pozitiv'
END AS EXPORT_IMPORT
FROM STATISTICI;
```

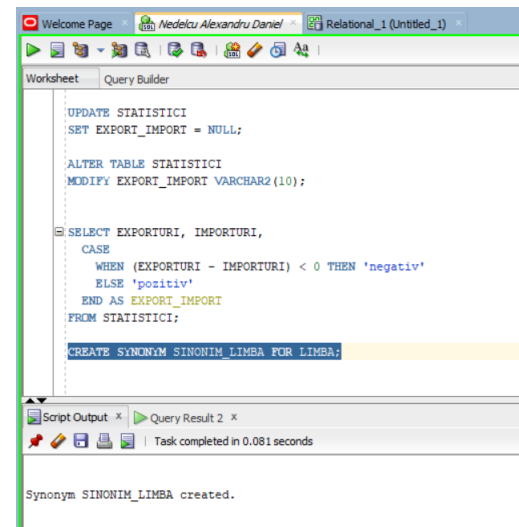
Below the query window, the 'Query Result' tab displays the following data:

	EXPORTURI	IMPORTURI	EXPORT_IMPORT
1	421	321	pozitiv
2	2421	3231	negativ
3	151	26	pozitiv
4	9087	5321	pozitiv
5	21	31	negativ
6	51421	52321	negativ
7	6511	8321	negativ
8	21	21	pozitiv
9	5151	6321	negativ

Gestiunea altor obiecte ale bazei de date: vederi, indcsi, sinonime, secvente.

Sa se creeze un sinonim pentru tabela LIMBA.

```
CREATE SYNONYM SINONIM_LIMBA FOR LIMBA;
```



The screenshot shows the SQL Developer interface with a query window containing the following SQL code:

```
UPDATE STATISTICI
SET EXPORT_IMPORT = NULL;

ALTER TABLE STATISTICI
MODIFY EXPORT_IMPORT VARCHAR2(10);

SELECT EXPORTURI, IMPORTURI,
CASE
  WHEN (EXPORTURI - IMPORTURI) < 0 THEN 'negativ'
  ELSE 'pozitiv'
END AS EXPORT_IMPORT
FROM STATISTICI;

CREATE SYNONYM SINONIM_LIMBA FOR LIMBA;
```

Below the query window, the 'Query Result' tab displays the following message:

Synonym SINONIM_LIMBA created.

Sa se creeze o vedere cu toate tarile care nu incep cu litera A.

CREATE OR REPLACE VIEW VEDERE_TARI_FARA_A AS

SELECT *

FROM TARA

WHERE UPPER(SUBSTR(NUME_TARA, 1, 1)) NOT IN ('A');

The screenshot displays a database management interface with the following components:

- Connections:** A tree view on the left showing the database structure, including tables like `VEDERE_TARI_FARA_A`, `ID_TARA`, `NUME_TARA`, `CAPITALA`, `AN_ADERARE`, and `ID_CONTINENT`.
- Worksheet:** The central area showing the SQL script for creating the view `VEDERE_TARI_FARA_A`. The script includes a comment `END AS EXPORT_IMPORT`, a `FROM STATISTICI;` statement, and the `CREATE OR REPLACE VIEW` statement with the `SELECT *` and `WHERE` clause.
- Script Output:** A message indicating "Task completed in 0.042 seconds".
- Reports:** A section on the left with various report types like Analytic View Reports, Data Dictionary Reports, etc.
- Browser:** A section on the left with various browser types like Dimensions, Directories, etc.
- View Results:** A table at the bottom showing the data for the `VEDERE_TARI_FARA_A` view. The table has columns `ID_TARA`, `NUME_TARA`, `CAPITALA`, `AN_ADERARE`, and `ID_CONTINENT`. The data is as follows:

ID_TARA	NUME_TARA	CAPITALA	AN_ADERARE	ID_CONTINENT
1	Romania	Bucuresti	2005	1
2	Danemarca	Copenhaga	2005	1
3	Spania	Madrid	2008	1
4	Franta	Paris	2006	1
5	SUA	Washington DC	2021	4
6	India	New Delhi	2017	2
7	Yemen	Sanaa	2014	2
8	Rusia	Moscova	2023	9
9	Egipt	Cairo	2010	5