PROIECT

BAZE DE DATE

PROFESOR : NUME STUDENT :

CORBEA ALEXANDRA MARIA MARINESCU DANA MARIA

GRUPA : 1045

CUPRINS

1. Descrierea temei economice alese
2. Schema conceptuală pentru modelarea temei alese
3. Construirea bazei de date – tabele, legături între tabele şi restricţii de integritate. Exemplificarea operaţiilor LDD
4. Exemple cu operaţii de actualizare a datelor: INSERT, UPDATE, DELETE
5. Exemple de interogări cât mai variate şi relevante pentru tema aleasă

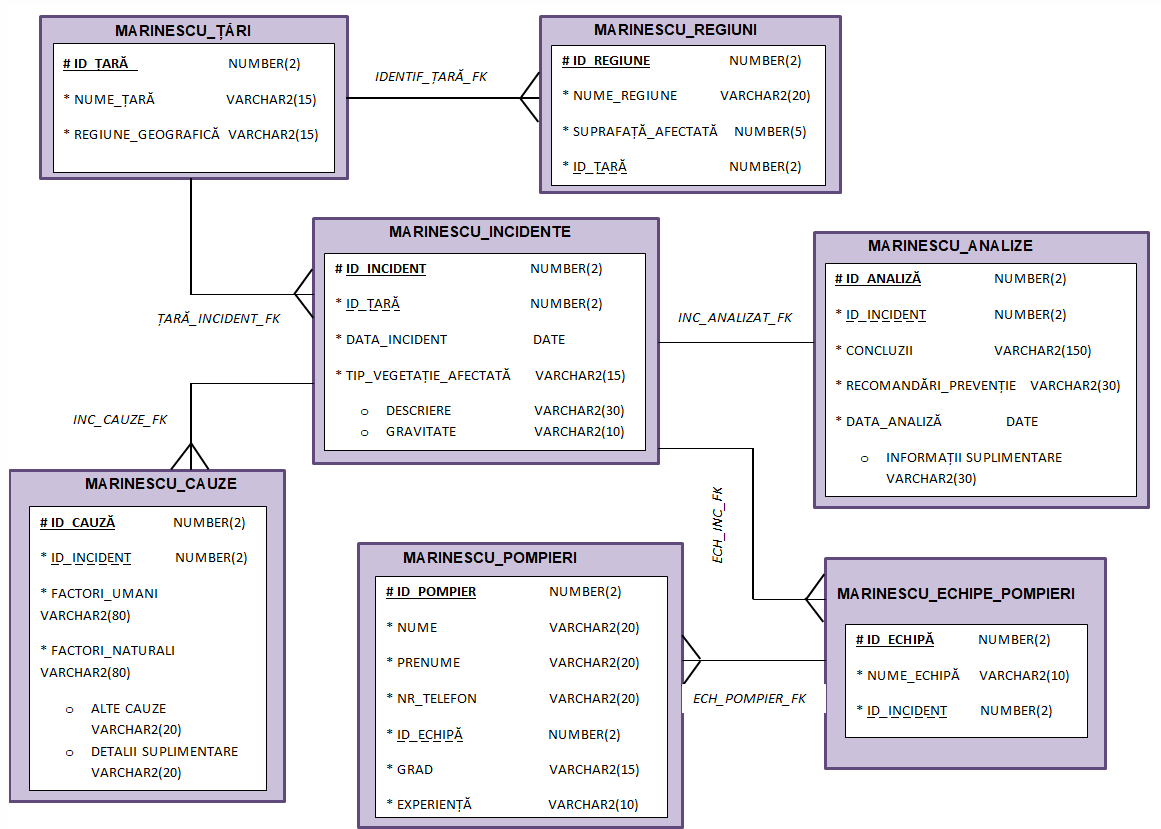
1. Descrierea temei economice alese

Tema economică aleasă pentru această bază de date este dată de evidența **incendiilor forestiere** din întreaga lume, având implicații semnificative în gestionarea resurselor, protejarea mediului și asigurarea securității populației.

Această bază de date are scopul de a colecta, stoca și gestiona informații legate de incendiile forestiere din toată lumea. Ea oferă o perspectivă globală asupra incendiilor forestiere, furnizând date relevante pentru autoritățile guvernamentale, organizațiile non-guvernamentale, cercetători și comunități afectate.

Baza de date conține informații despre incidentele ce au avut loc în ultimii 30 de ani precum locația, suprafața și tipul de vegetație afectată, gravitatea incendiului, dar și despre echipajele de pompieri ce au participat la oprirea incendiilor. Totodată, aceste informații sunt sintetizate într-o tabelă ce poartă numele de Analize, întrunind informațiile anterioare într-un raport complet, incluzând și măsuri de prevenție ale acestora.



2. Schema conceptuală pentru modelarea temei alese

3. Construirea bazei de date – tabele, legături între tabele şi restricţii de integritate. Exemplificarea operaţiile LDD

CREARE TABELE

- COMANDA **CREATE**

*Să se creeze tabelele pentru entitățile bazei de date.*

* CREATE TABLE **MARINESCU\_POMPIERI**

(

ID\_POMPIER NUMBER(2) PRIMARY KEY,

NUME VARCHAR2(20),

PRENUME VARCHAR2(20),

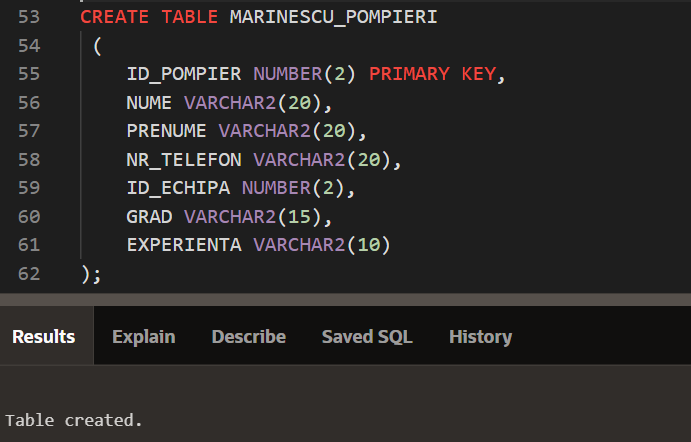
NR\_TELEFON VARCHAR2(20),

ID\_ECHIPA NUMBER(2),

GRAD VARCHAR2(15),

EXPERIENTA VARCHAR2(10)

);



* CREATE TABLE **MARINESCU\_INCIDENTE**

(

ID\_INCIDENT NUMBER(2) PRIMARY KEY,

ID\_TARA NUMBER(2),

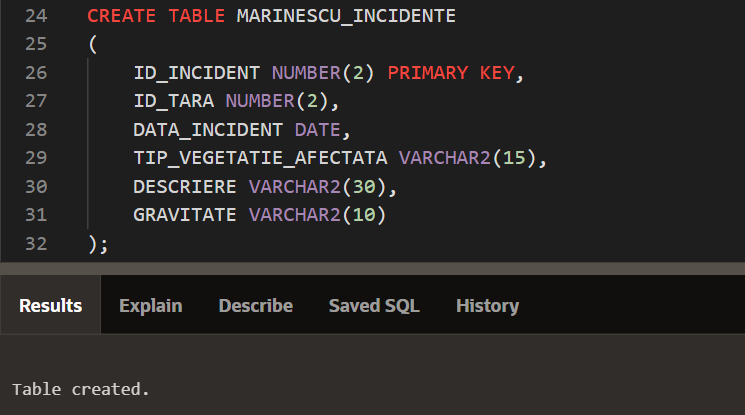
DATA\_INCIDENT DATE,

TIP\_VEGETATIE\_AFECTATA VARCHAR2(15),

DESCRIERE VARCHAR2(30),

GRAVITATE VARCHAR2(10)

);



CREATE TABLE **MARINESCU\_CAUZE**

(

ID\_CAUZA NUMBER(2) PRIMARY KEY,

ID\_INCIDENT NUMBER(2),

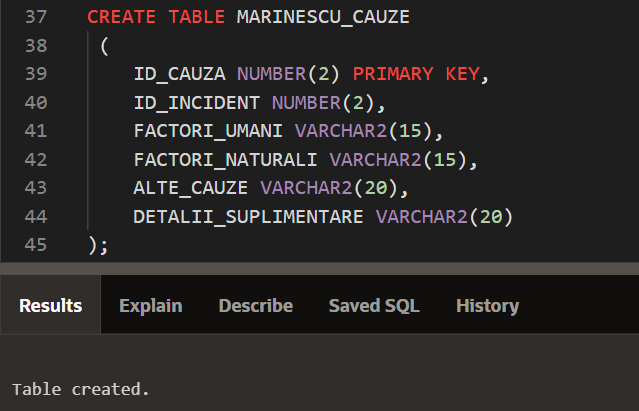
FACTORI\_UMANI VARCHAR2(15),

FACTORI\_NATURALI VARCHAR2(15),

ALTE\_CAUZE VARCHAR2(20),

DETALII\_SUPLIMENTARE VARCHAR2(20)

);



- COMANDA **ALTER**

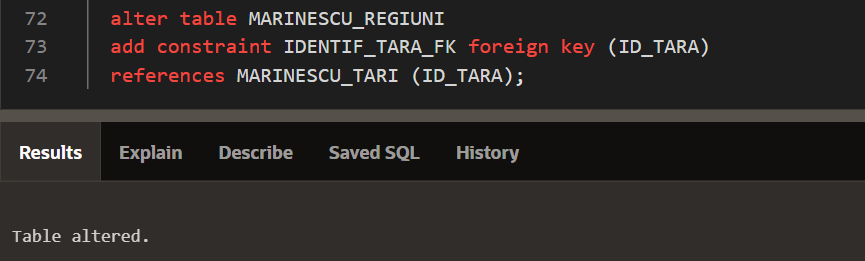
ADĂUGARE RESTRICȚII DE INTEGRITATE

- *Să se adauge chei externe, reprezentând legăturile dintre tabele.*

* alter table MARINESCU\_REGIUNI

add constraint IDENTIF\_TARA\_FK foreign key (ID\_TARA)

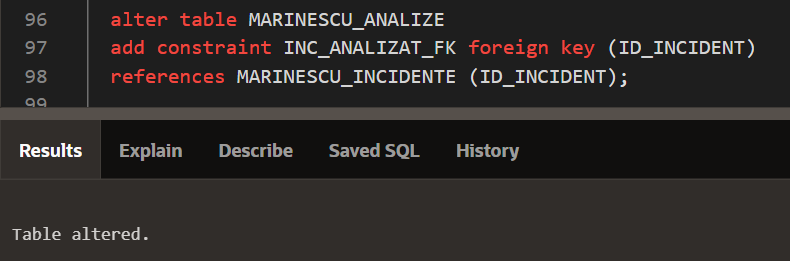
references MARINESCU\_TARI (ID\_TARA);



* alter table MARINESCU\_ANALIZE

add constraint INC\_ANALIZAT\_FK foreign key (ID\_INCIDENT)

references MARINESCU\_INCIDENTE (ID\_INCIDENT);

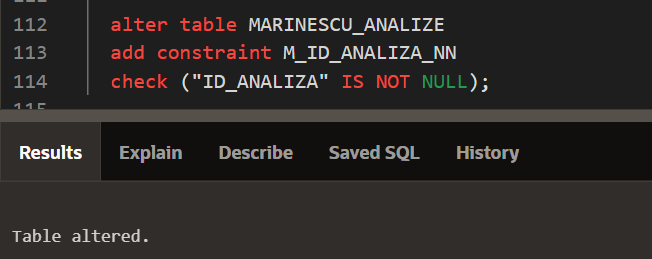


- *Să se adauge restricția de obligativitate a introducerii unei valori pentru coloana ID\_ANALIZA din tabela MARINESCU\_ANALIZE.*

* alter table MARINESCU\_ANALIZE

add constraint M\_ID\_ANALIZA\_NN

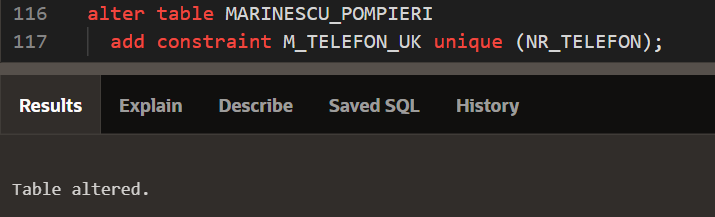
check ("ID\_ANALIZA" IS NOT NULL);



- *Să se adauge restricția de unicitate pentru coloana NR\_TELEFON din tabela MARINESCU\_POMPIERI.*

* alter table MARINESCU\_POMPIERI

add constraint M\_TELEFON\_UK unique (NR\_TELEFON);

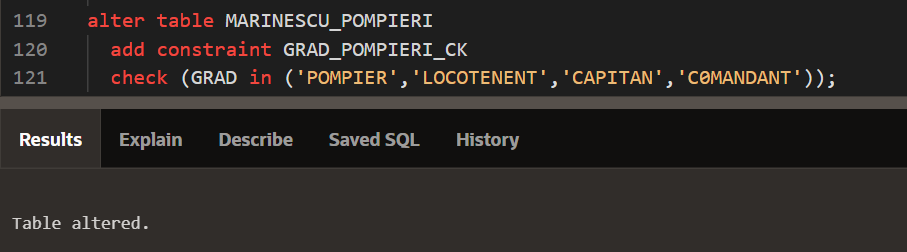


- *Să se adauge restricția de comportament pentru coloana GRAD din tabela MARINESCU\_POMPIERI.*

* alter table MARINESCU\_POMPIERI

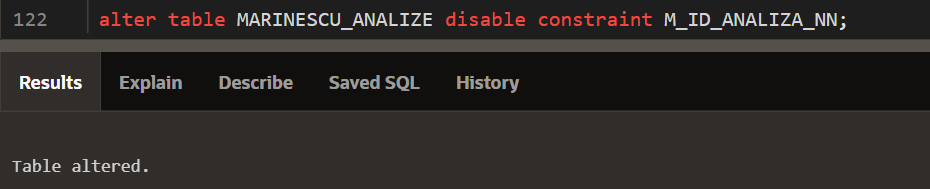
add constraint GRAD\_POMPIERI\_CK

check (GRAD in ('POMPIER','LOCOTENENT','CAPITAN','COMANDANT'));



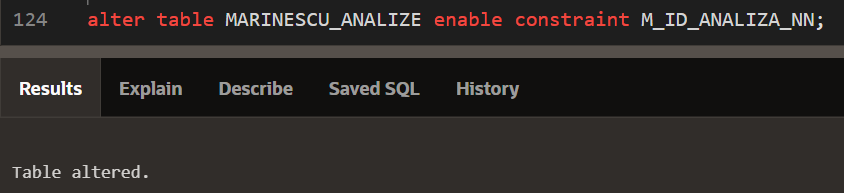
- *Să se dezactiveze restricția M\_ID\_ANALIZA\_NN.*

* alter table MARINESCU\_ANALIZE disable constraint M\_ID\_ANALIZA\_NN;



- *Să se activeze restricția M\_ID\_ANALIZA\_NN*.

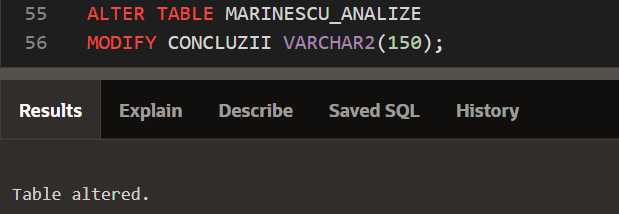
* alter table MARINESCU\_ANALIZE enable constraint M\_ID\_ANALIZA\_NN;

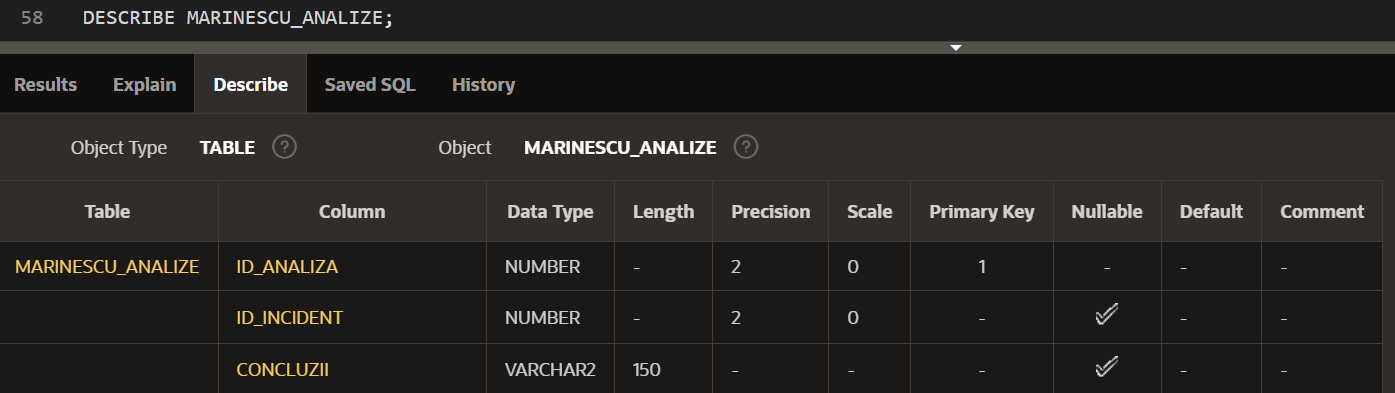


- *Să se modifice tipul de dată al coloanei CONCLUZII din tabela MARINESCU\_ANALIZE.*

ALTER TABLE MARINESCU\_ANALIZE

MODIFY CONCLUZII VARCHAR2(150);

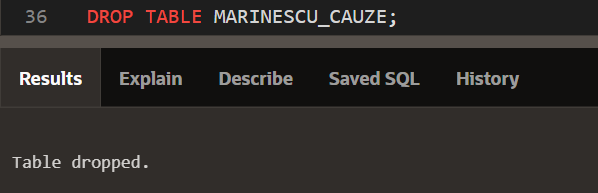




- COMANDA **DROP**

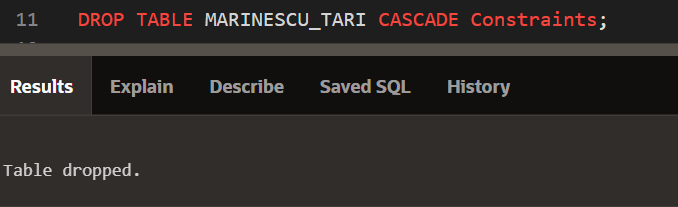
- *Să se șteargă tabela MARINESCU\_CAUZE.*

* DROP TABLE MARINESCU\_CAUZE;



- *Să se șteargă tabela MARINESCU\_TARI și restricțiile asociate.*

* DROP TABLE MARINESCU\_TARI CASCADE Constraints;



4. Exemple cu operaţii de actualizare a datelor: **INSERT, UPDATE, DELETE**

**-** COMANDA **INSERT**

*Să se adauge înregistrări în tabele.*

* INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (1,'ITALIA','EUROPA');

INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (2,'ROMANIA','EUROPA');

INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (3,'PORTUGALIA','EUROPA');

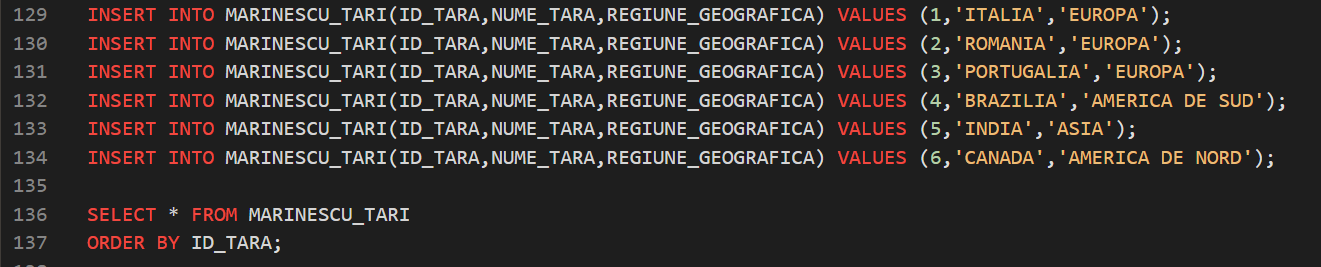
INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (4,'BRAZILIA','AMERICA DE SUD');

INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (5,'INDIA','ASIA');

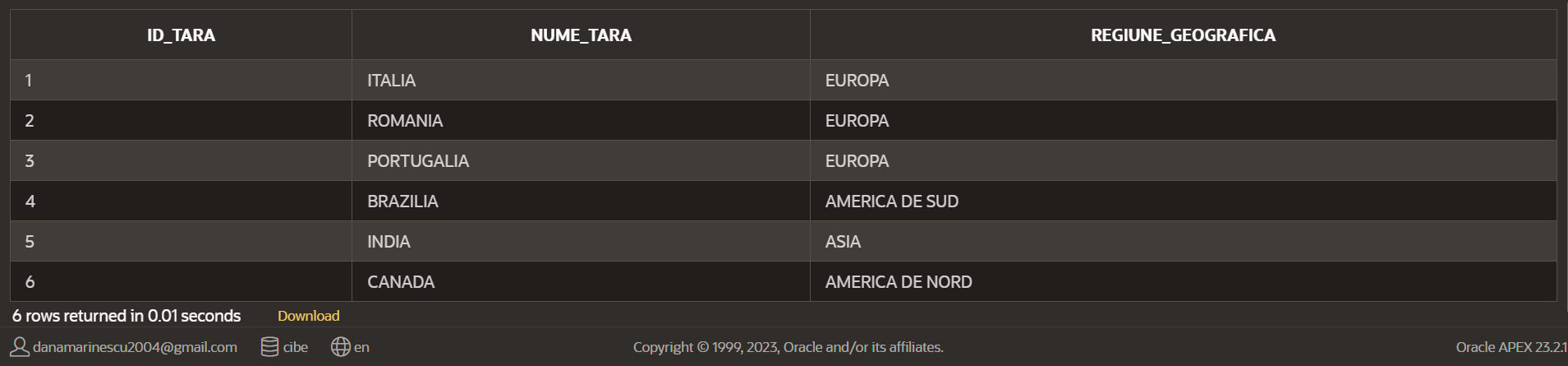
INSERT INTO MARINESCU\_TARI(ID\_TARA,NUME\_TARA,REGIUNE\_GEOGRAFICA) VALUES (6,'CANADA','AMERICA DE NORD');

SELECT \* FROM MARINESCU\_TARI

ORDER BY ID\_TARA;



REZULTAT



* INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA ,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

(1,1,TO\_DATE('12-02-2000','DD-MM-YYYY'),'CONIFERE','-','REDUSA');

INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA

,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

(2,4,TO\_DATE('15-06-1998','DD-MM-YYYY'),'CONIFERE','-','REDUSA');

INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA ,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

(3,2,TO\_DATE('22-04-2005','DD-MM-YYYY'),'FOIOASE','-','RIDICATA');

INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA ,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

(4,5,TO\_DATE('16-08-1994','DD-MM-YYYY'),'CONIFERE','-','RIDICATA');

INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA

,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

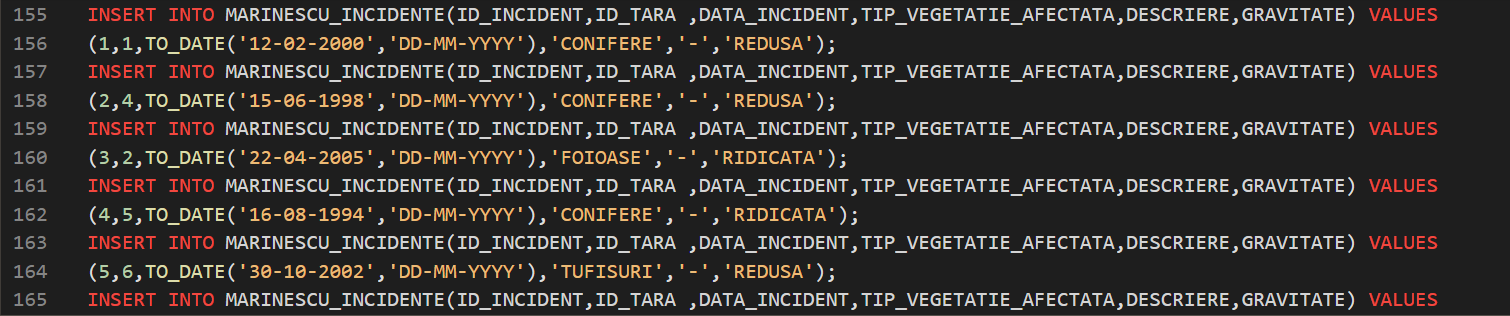
(5,6,TO\_DATE('30-10-2002','DD-MM-YYYY'),'TUFISURI','-','REDUSA');

INSERT INTO MARINESCU\_INCIDENTE(ID\_INCIDENT,ID\_TARA ,DATA\_INCIDENT,TIP\_VEGETATIE\_AFECTATA,DESCRIERE,GRAVITATE) VALUES

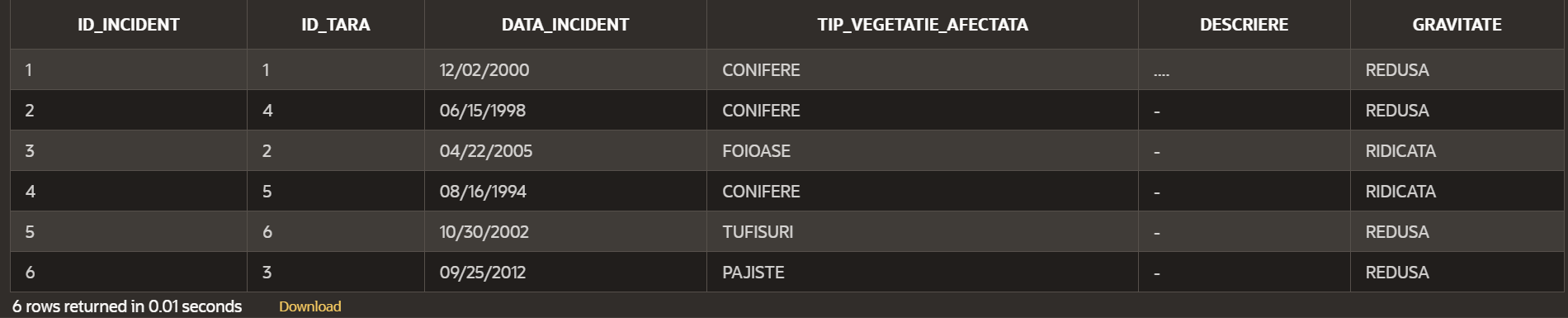
(6,3,TO\_DATE('25-09-2012','DD-MM-YYYY'),'PAJISTE','-','REDUSA');

SELECT \* FROM MARINESCU\_INCIDENTE

ORDER BY ID\_INCIDENT;



REZULTAT



* INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

(1,'POPESCU','ANDREI','0723456789',2,'POMPIER','2 ANI');

INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

(2,'MARINESCU','DANA','0773459829',1,'CAPITAN','5 ANI');

INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

(3,'DADU','ALEXANDRA','0765453489',3,'LOCOTENENT','3 ANI');

INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

(4,'BRATU','CRISTIAN','0712356799',6,'COMANDANT','10 ANI');

INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

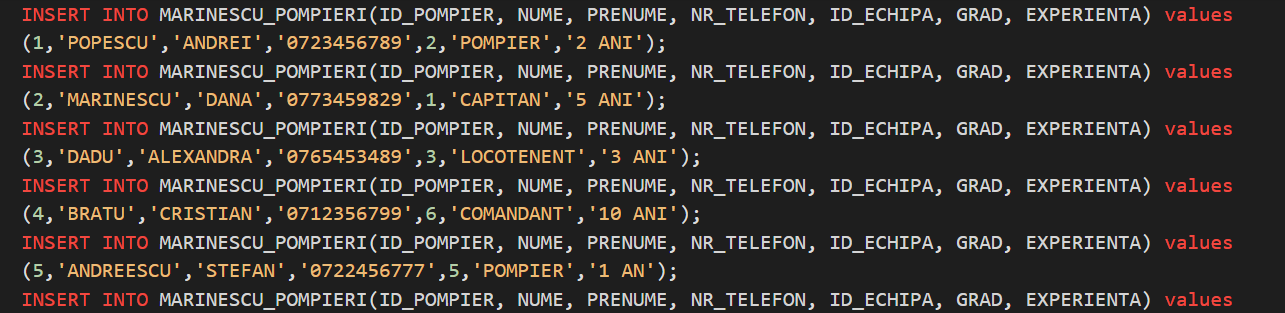
(5,'ANDREESCU','STEFAN','0722456777',5,'POMPIER','1 AN');

INSERT INTO MARINESCU\_POMPIERI(ID\_POMPIER, NUME, PRENUME, NR\_TELEFON, ID\_ECHIPA, GRAD, EXPERIENTA) values

(6,'NITU','ALEXIA','0723433389',4,'LOCOTENENT','3 ANI');

SELECT \* FROM MARINESCU\_POMPIERI

ORDER BY ID\_POMPIER;



REZULTAT



- COMANDA **UPDATE**

- *Să se actualizeze coloana CONCLUZII din tabela MARINESCU\_ANALIZE , astfel : unde apare “REDUSA” in coloana gravitate din tabela MARINESCU\_INCIDENTE , sa se scrie ‘’INCENDIU DE GRAD redus”, iar unde apare ‘’RIDICATA’’, să se scrie ‘’INCENDIU DE GRAD RIDICAT”.*

* **UPDATE** MARINESCU\_ANALIZE

SET CONCLUZII =

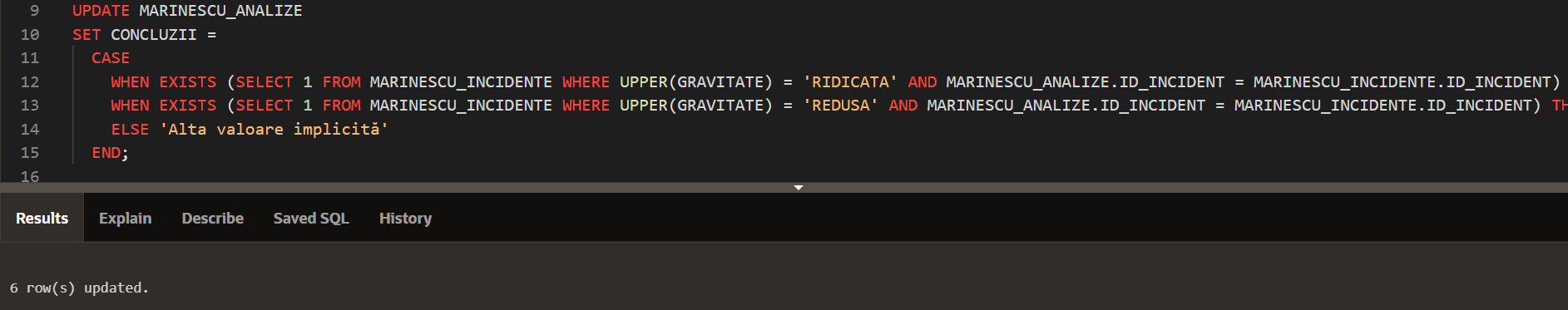
CASE

WHEN EXISTS (SELECT 1 FROM MARINESCU\_INCIDENTE WHERE UPPER(GRAVITATE) = 'RIDICATA' AND MARINESCU\_ANALIZE.ID\_INCIDENT = MARINESCU\_INCIDENTE.ID\_INCIDENT) THEN 'INCENDIU DE GRAD RIDICAT'

WHEN EXISTS (SELECT 1 FROM MARINESCU\_INCIDENTE WHERE UPPER(GRAVITATE) = 'REDUSA' AND MARINESCU\_ANALIZE.ID\_INCIDENT = MARINESCU\_INCIDENTE.ID\_INCIDENT) THEN 'INCENDIU DE GRAD REDUS'

ELSE 'Alta valoare implicită'

END;



REZULTAT



*- Să se actualizeze coloana RECOMANDARI\_PREVENTIE din tabela MARINESCU\_ANALIZE , astfel : unde incendiul este cauzat de FACTORI NATURALI ( în tabela MARINESCU\_CAUZE ) , să se scrie ‘’MONITORIZARE METEOROLOGICĂ”, iar unde incendiul este cauzat de FACTORI UMANI, să se scrie ‘’BARIERE DE PROTECȚIE”*

* UPDATE MARINESCU\_ANALIZE

SET RECOMANDARI\_PREVENTIE =

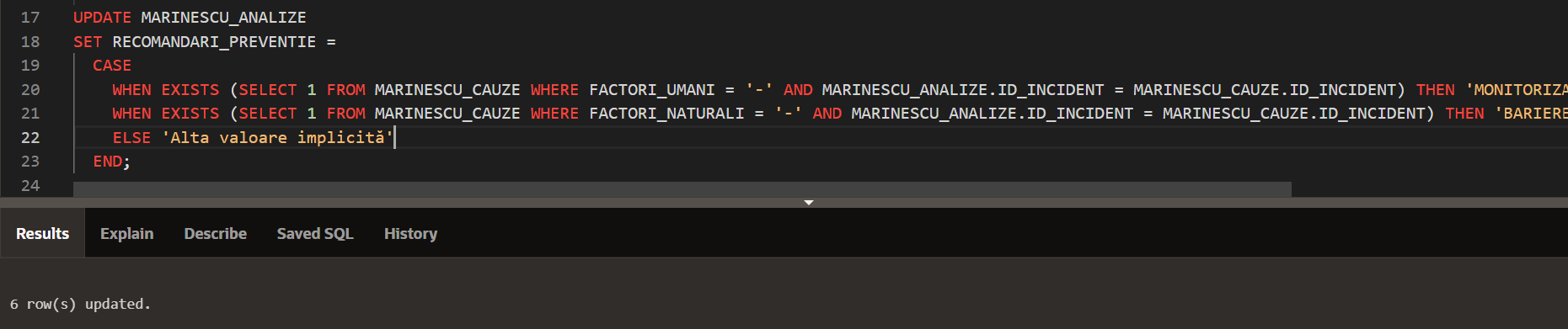
CASE

WHEN EXISTS (SELECT 1 FROM MARINESCU\_CAUZE WHERE FACTORI\_UMANI = '-' AND MARINESCU\_ANALIZE.ID\_INCIDENT = MARINESCU\_CAUZE.ID\_INCIDENT) THEN 'MONITORIZARE METEOROLOGICA'

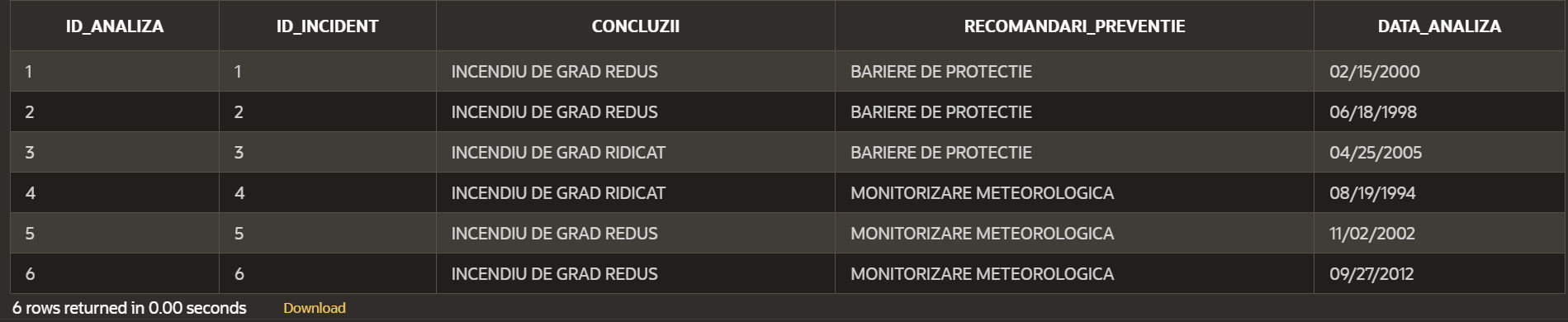
WHEN EXISTS (SELECT 1 FROM MARINESCU\_CAUZE WHERE FACTORI\_NATURALI = '-' AND MARINESCU\_ANALIZE.ID\_INCIDENT = MARINESCU\_CAUZE.ID\_INCIDENT) THEN 'BARIERE DE PROTECTIE'

ELSE 'Alta valoare implicită'

END;



REZULTAT



- *Să se actualizeze coloana NUME\_ECHIPA din tabela MARINESCU\_ECHIPE\_POMPIERI, astfel : numele echipei cu ID=1 , va deveni ‘’AVENTURA’’.*

* UPDATE MARINESCU\_ECHIPE\_POMPIERI

SET NUME\_ECHIPA='AVENTURA'

WHERE ID\_ECHIPA=1;

SELECT \* FROM MARINESCU\_ECHIPE\_POMPIERI

ORDER BY ID\_ECHIPA;

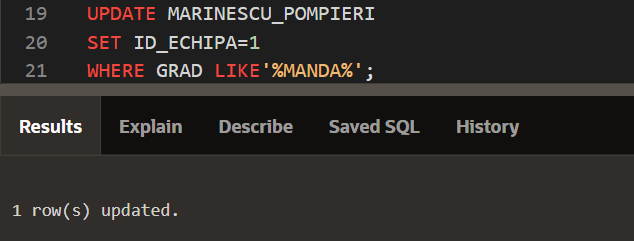


- *Să se actualizeze coloana ID\_ECHIPA din tabela MARINESCU\_POMPIERI, astfel : unde se regăsește textul ‘’MANDA’’ în funcțiile din coloana GRAD, ID-ul echipei va deveni 1. ( țoți comandanții vor face parte din echipa 1 ) .*

* UPDATE MARINESCU\_POMPIERI

SET ID\_ECHIPA=1

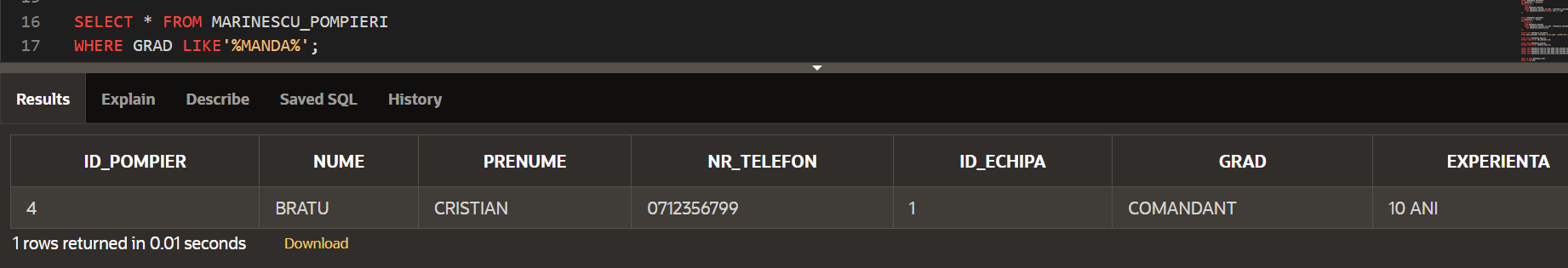
WHERE GRAD LIKE'%MANDA%';



REZULTAT

SELECT \* FROM MARINESCU\_POMPIERI

WHERE GRAD LIKE'%MANDA%';



- *Să se adauge o nouă coloană în tabela MARINESCU\_POMPIERI , cu numele ID\_SUPERIOR, de tipul NUMBER(2).*

ALTER TABLE MARINESCU\_POMPIERI

ADD ID\_SUPERIOR NUMBER(2);

- *Să se adauge valori în coloana ID\_SUPERIOR din tabela MARINESCU\_POMPIERI, astfel :*

* *unde* ***ID\_POMPIER=2*** *( ocupă poziția de Căpitan ) , ID\_SUPERIOR=4 ( Comandantul este singurul superior al căpitanului )*
* *unde* ***ID\_POMPIER=1,3,5,6*** *( Pompieri ) , ID\_SUPERIOR=2 ( Restul pompierilor se subordonează căpitanului )*
* UPDATE MARINESCU\_POMPIERI

SET ID\_SUPERIOR =

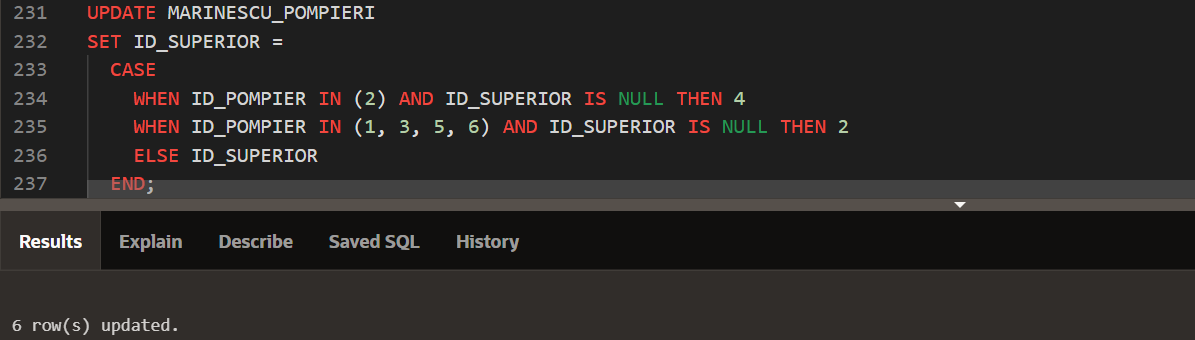
CASE

WHEN ID\_POMPIER **IN** (2) AND ID\_SUPERIOR **IS NULL** THEN 4

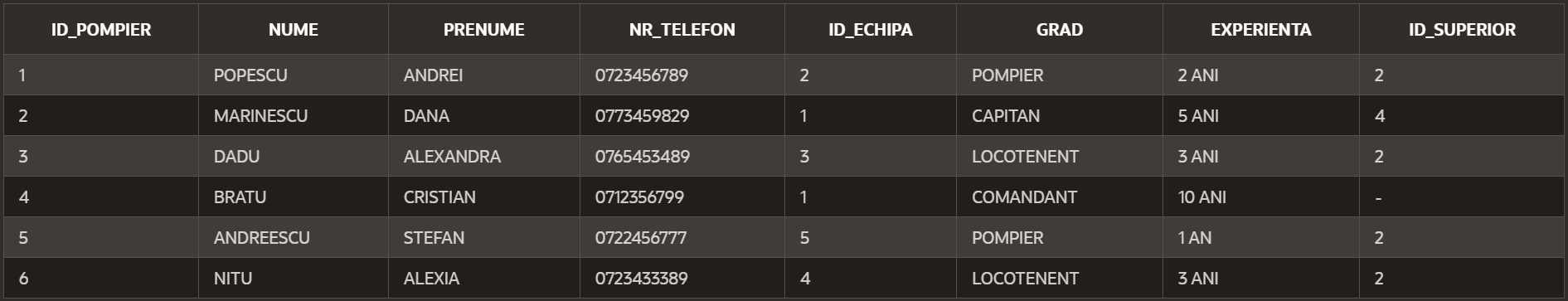
WHEN ID\_POMPIER **IN** (1, 3, 5, 6) AND ID\_SUPERIOR **IS NULL** THEN 2

ELSE ID\_SUPERIOR

END;

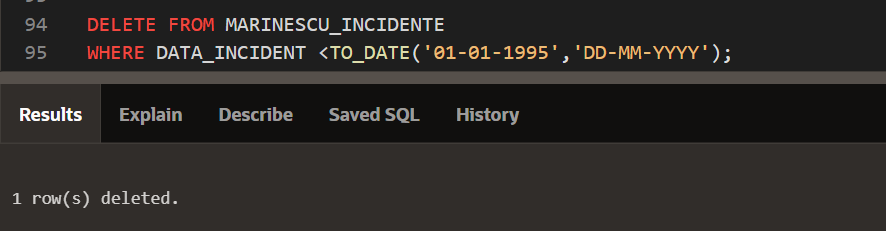


REZULTAT

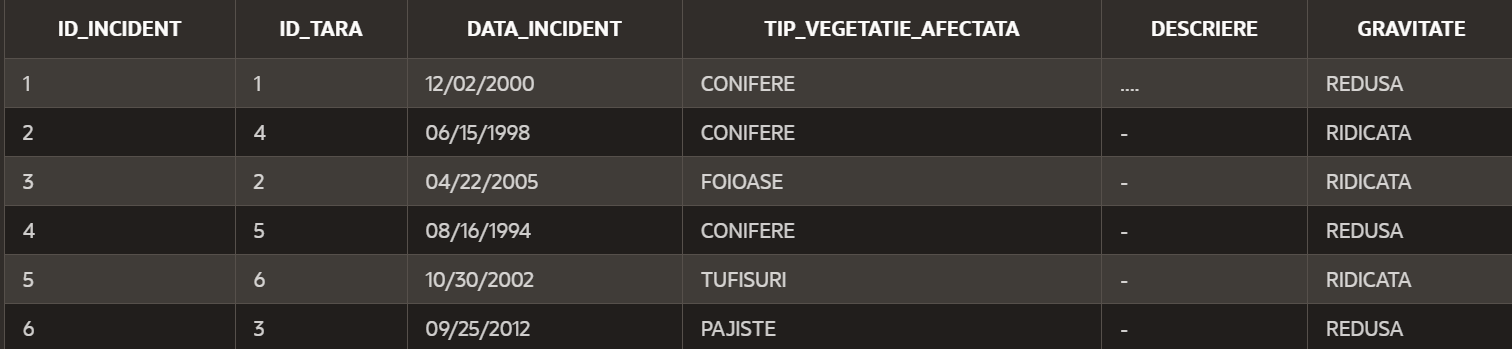


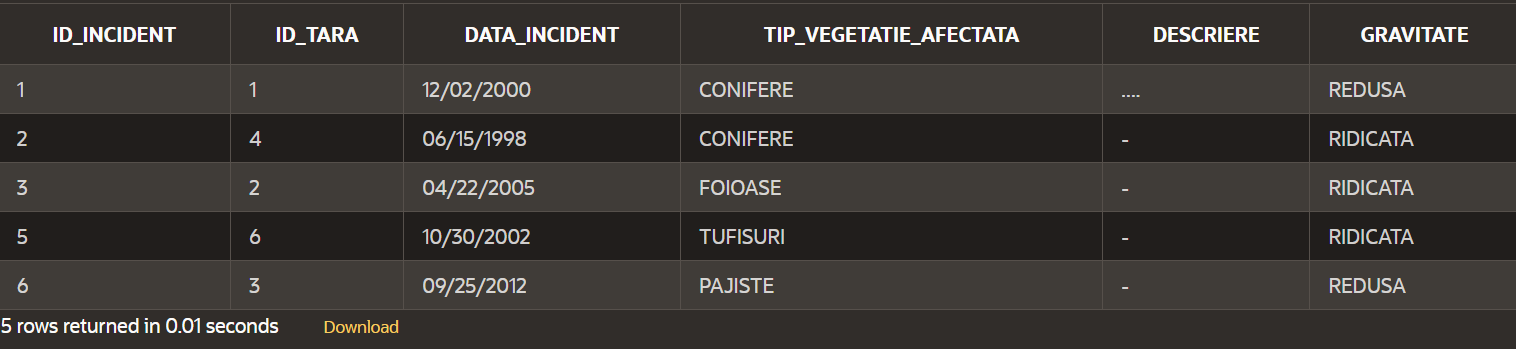
- COMANDA **DELETE**

*- Să se șteargă din tabela MARINESCU\_INCIDENTE înregistrările ce au avut loc înainte de 1.01.1995.*



ÎNAINTE

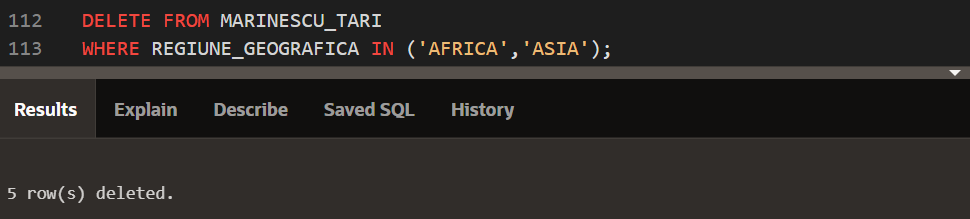


DUPĂ

*- Să se șteargă din tabela MARINESCU\_TARI înregistrările ce conțin tări care fac parte din continentele Africa sau Asia.*

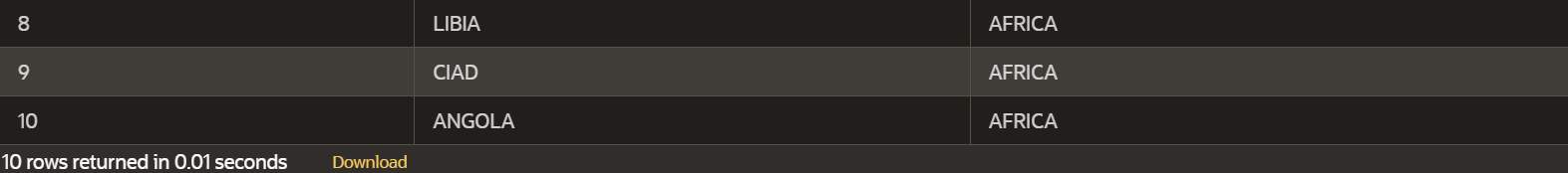
* DELETE FROM MARINESCU\_TARI

WHERE REGIUNE\_GEOGRAFICA IN ('AFRICA','ASIA');

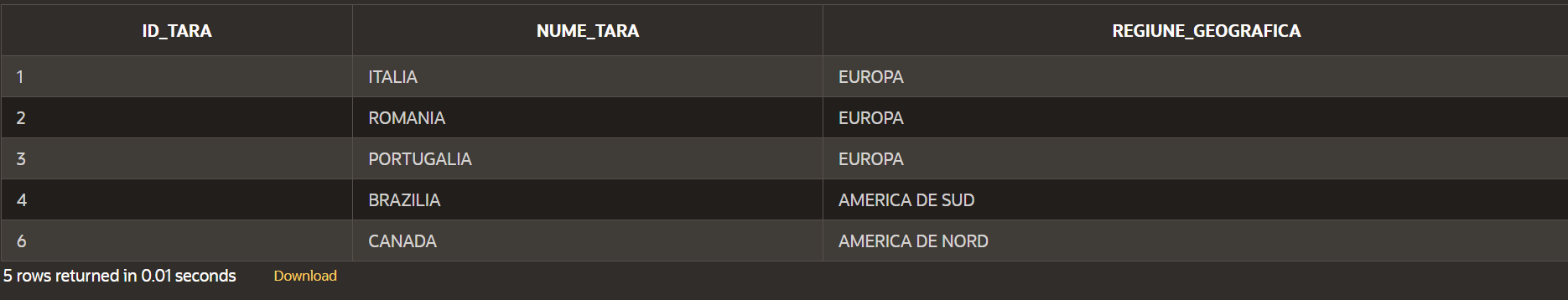


ÎNAINTE





DUPĂ



5. Exemple de **interogări** cât mai variate şi relevante pentru tema aleasă

- *JONCTIUNE DE EGALITATE*

*Să se realizeze o joncțiune de egalitate ( se selectează rândurile unde există o potrivire între valorile coloanei ID\_TARA din ambele tabele ) între tabelele MARINESCU\_INCIDENTE și MARINESCU\_REGIUNI .*

SELECT MI.\*,SUPRAFATA\_AFECTATA

FROM MARINESCU\_INCIDENTE MI , MARINESCU\_REGIUNI MR

WHERE MI.ID\_TARA**=**MR.ID\_TARA

ORDER BY ID\_INCIDENT;



- *Să se modifice coloana GRAVITATE din tabela MARINESCU\_INCIDENTE , să ia valoarea expresiei ‘’RIDICATA’’ pentru incidentele cu suprafata afectata ( din tabela MARINESCU\_REGIUNI ) între 400 și 1200 ha.*

* UPDATE MARINESCU\_INCIDENTE

SET GRAVITATE = 'RIDICATA'

WHERE EXISTS

(

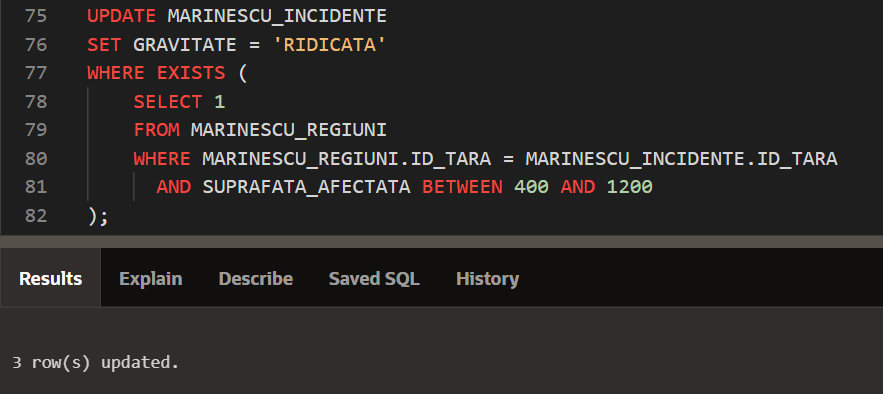
SELECT 1

FROM MARINESCU\_REGIUNI

WHERE MARINESCU\_REGIUNI.ID\_TARA **=** MARINESCU\_INCIDENTE.ID\_TARA

AND SUPRAFATA\_AFECTATA **BETWEEN 400 AND 1200**

);



REZULTAT



- *Să se modifice coloana GRAVITATE din tabela MARINESCU\_INCIDENTE , să ia valoarea expresiei ‘’REDUSA’’ pentru incidentele cu suprafața afectată ( din tabela MARINESCU\_REGIUNI ) mai mică de 350 ha.*

* UPDATE MARINESCU\_INCIDENTE

SET GRAVITATE = 'REDUSA'

WHERE EXISTS (

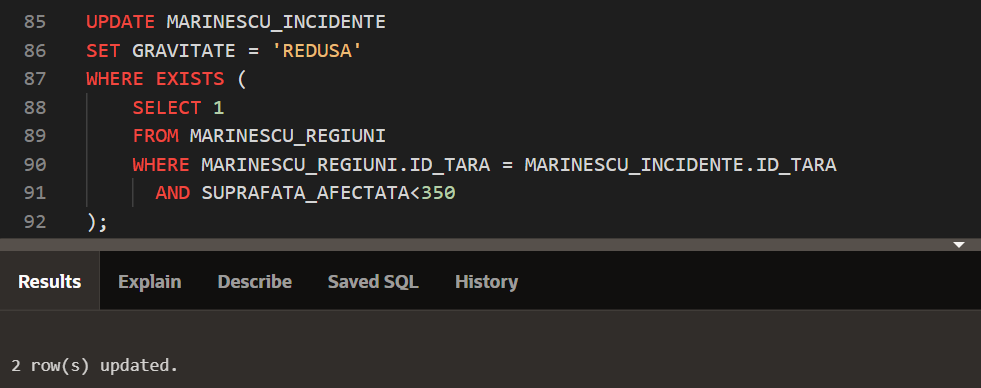
SELECT 1

FROM MARINESCU\_REGIUNI

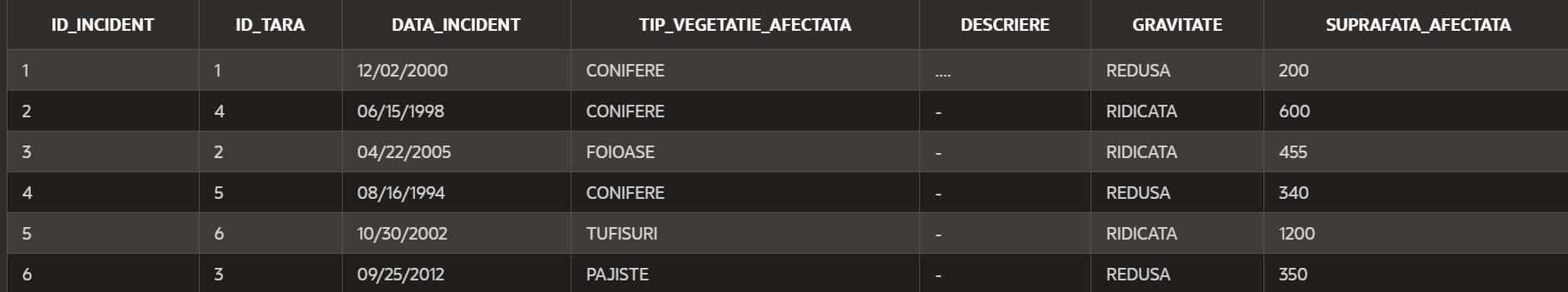
WHERE MARINESCU\_REGIUNI.ID\_TARA = MARINESCU\_INCIDENTE.ID\_TARA

AND SUPRAFATA\_AFECTATA<350

);



REZULTAT



- *Să se utilizeze funcția EXTRACT pentru a obține anul și luna din coloana DATA\_INCIDENT din tabela MARINESCU\_INCIDENTE și să se realizeze o joncțiune( de tip INNER JOIN ) între MARINESCU\_INCIDENTE și MARINESCU\_REGIUNI pe baza coloanei ID\_TARA.*

SELECT

EXTRACT(YEAR FROM MI.DATA\_INCIDENT) AS AN\_INCIDENT,

EXTRACT(MONTH FROM MI.DATA\_INCIDENT) AS LUNA\_INCIDENT,

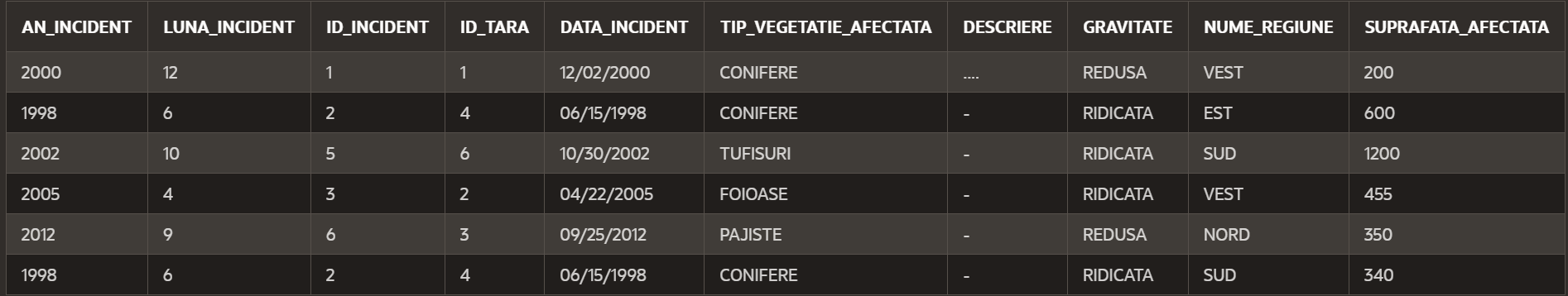
MI.\*, NUME\_REGIUNE, SUPRAFATA\_AFECTATA

FROM

MARINESCU\_INCIDENTE MI

JOIN

MARINESCU\_REGIUNI MR ON MI.ID\_TARA = MR.ID\_TARA;



- *Să se utilizeze funcția EXTRACT pentru a obține anul și luna din coloana DATA\_INCIDENT din tabela MARINESCU\_INCIDENTE și să se realizeze o joncțiune( de tip OUTER JOIN ) între MARINESCU\_INCIDENTE și MARINESCU\_REGIUNI pe baza coloanei ID\_TARA.*

SELECT

EXTRACT(YEAR FROM MI.DATA\_INCIDENT) AS AN\_INCIDENT,

EXTRACT(MONTH FROM MI.DATA\_INCIDENT) AS LUNA\_INCIDENT,

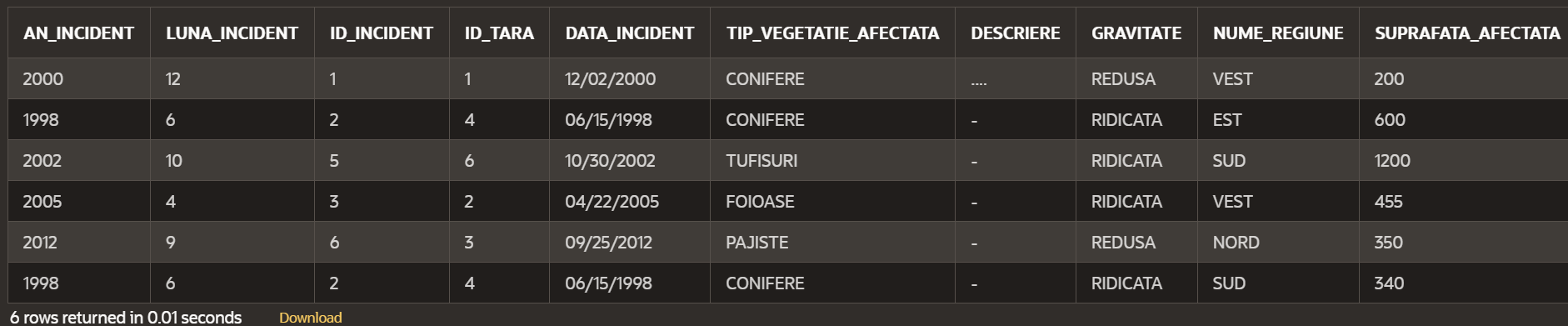
MI.\*, NUME\_REGIUNE, SUPRAFATA\_AFECTATA

FROM

MARINESCU\_INCIDENTE MI

LEFT OUTER JOIN

MARINESCU\_REGIUNI MR ON MI.ID\_TARA = MR.ID\_TARA;



*- Să se utilizeze funcția TO\_CHAR pentru a obține anul și luna din coloana DATA\_INCIDENT din tabela MARINESCU\_INCIDENTE sub formă de șiruri de caractere.*

* SELECT

TO\_CHAR(MI.DATA\_INCIDENT, 'YYYY') AS AN\_INCIDENT,

TO\_CHAR(MI.DATA\_INCIDENT, 'MM') AS LUNA\_INCIDENT,

MI.\*,

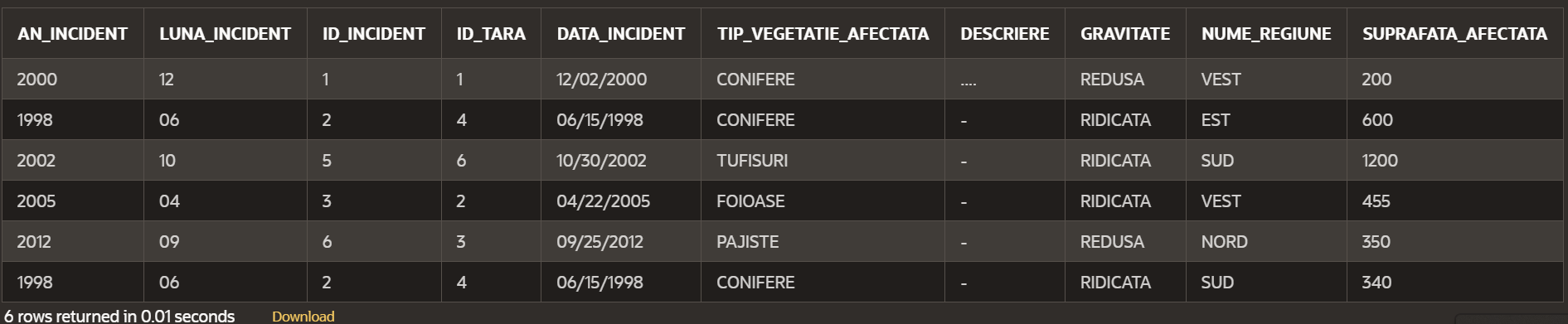
NUME\_REGIUNE,SUPRAFATA\_AFECTATA

FROM

MARINESCU\_INCIDENTE MI

JOIN

MARINESCU\_REGIUNI MR ON MI.ID\_TARA = MR.ID\_TARA;



- *Să se selecteze primele 2 caractere din numele țărilor din tabela MARINESCU\_TARI, ca PREFIX\_TARI.*

* SELECT SUBSTR(NUME\_TARA, 1, 2) AS PREFIX\_TARI

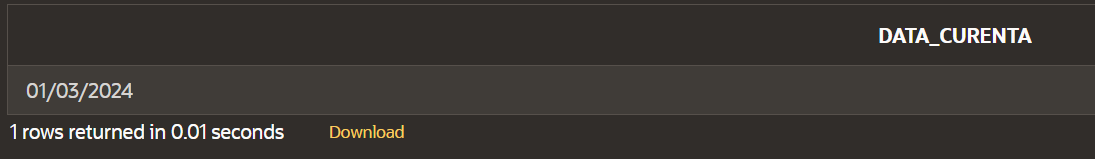
FROM MARINESCU\_TARI;



*- Să se afișeze data curentă.*

* SELECT SYSDATE AS DATA\_CURENTA

FROM DUAL;



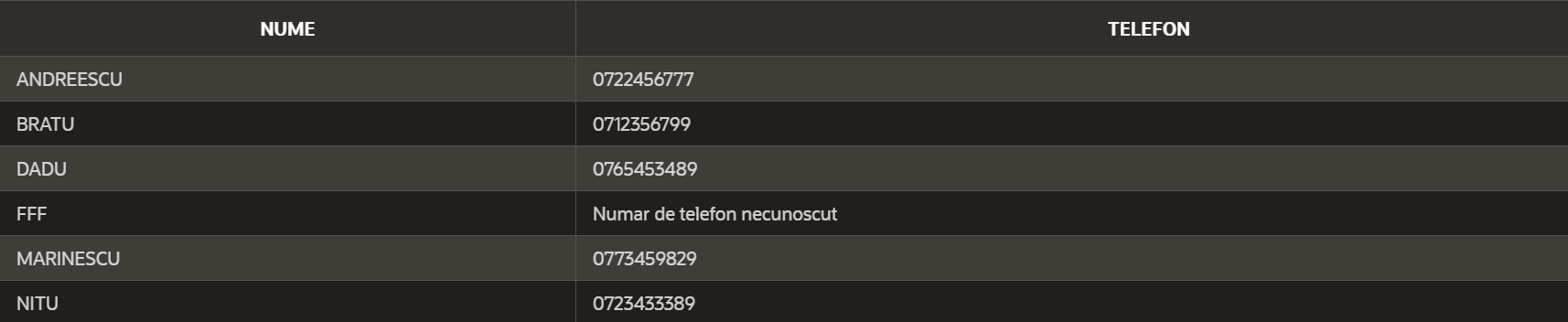
- *Să se utilizeze funcția* ***NVL*** *pentru a se verifica dacă există valori nule în coloana NR\_TELEFON din tabela MARINESCU\_POMPIERI, fiind ulterior înlocuite cu expresia ‘’Numar de telefon necunoscut’’.*

* SELECT

NUME,

NVL(NR\_TELEFON, 'Numar de telefon necunoscut') AS TELEFON

FROM MARINESCU\_POMPIERI;



*- Să se utilizeze funcția DECODE pentru a atribui un nivel corespunzător fiecărui grad.*

* SELECT

NUME,

PRENUME,

GRAD,

(SELECT DECODE(GRAD,

'POMPIER', 'Nivel de bază',

'LOCOTENENT', 'Nivel intermediar',

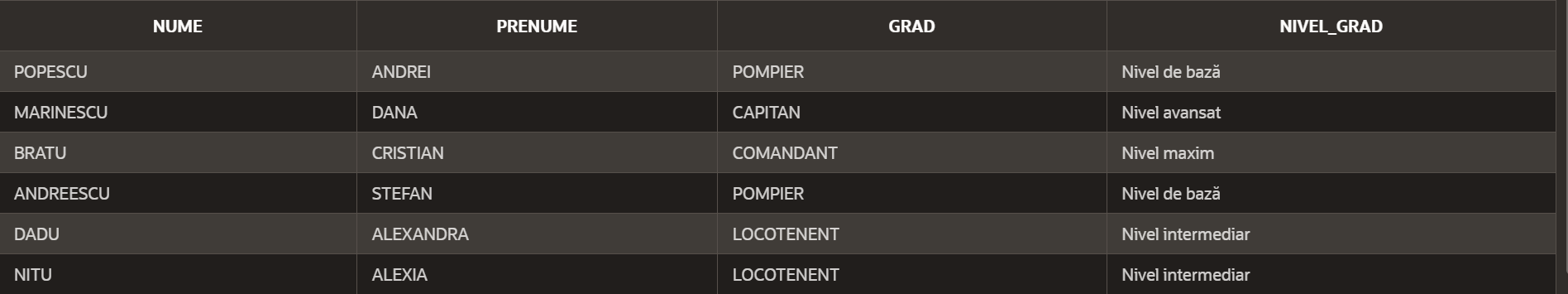
'CAPITAN', 'Nivel avansat',

'Nivel maxim') AS NIVEL\_GRAD

FROM DUAL) AS NIVEL\_GRAD

FROM

MARINESCU\_POMPIERI;

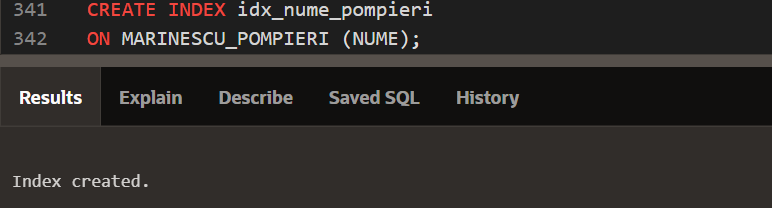


- *INDEXI, SINONIME, SECVENȚE*

*- Să se creeze un index pe baza tabelei MARINESCU\_POMPIERI.*

* CREATE INDEX idx\_nume\_pompieri

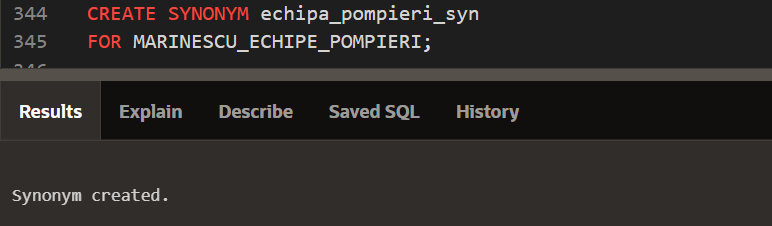
ON MARINESCU\_POMPIERI (NUME);



*- Să se creeze un sinonim pe baza tabelei MARINESCU\_ECHIPE\_POMPIERI.*

* CREATE SYNONYM echipa\_pompieri\_syn

FOR MARINESCU\_ECHIPE\_POMPIERI;

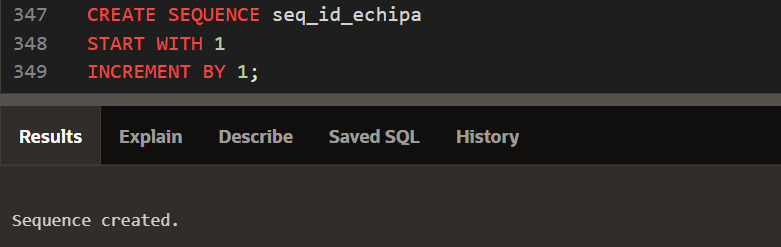


*- Să se creeze o secvență pe baza tabelei MARINESCU\_ECHIPE\_POMPIERI.*

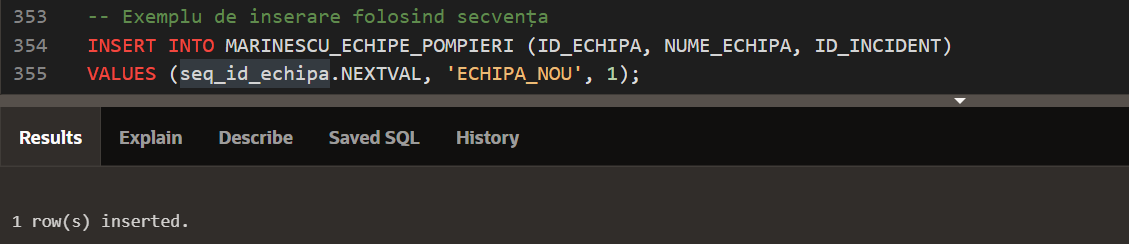
* CREATE SEQUENCE seq\_id\_echipa

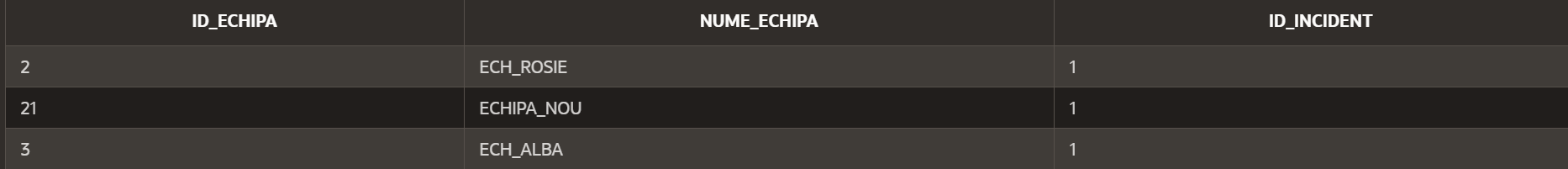
START WITH 1

INCREMENT BY 1;



- *EXEMPLU UTILIZARE SECVENȚĂ*





*- FUNCȚII DE GRUP*

*- Să se afișeze țările care au mai mult de 2 incidente.*

* SELECT

MI.ID\_TARA,

COUNT(MI.ID\_INCIDENT) AS NUMAR\_INCIDENTE

FROM

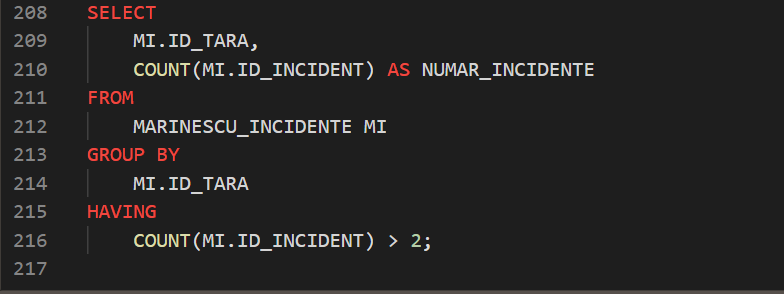
MARINESCU\_INCIDENTE MI

GROUP BY

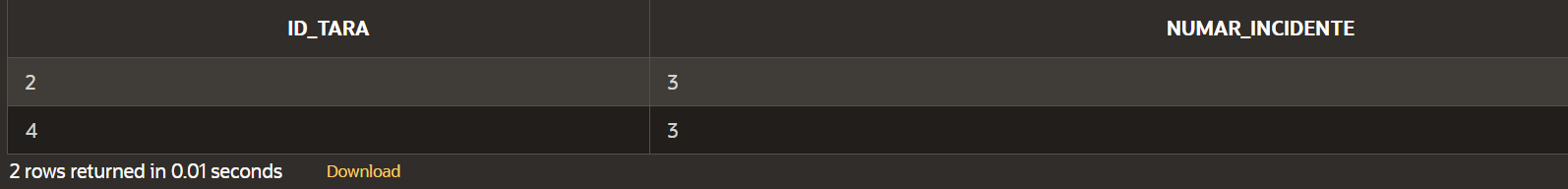
MI.ID\_TARA

HAVING

COUNT(MI.ID\_INCIDENT) > 2;



REZULTAT



*- CERERI IERARHICE*

*- Să se realizeze o reprezentare ierarhică a datelor din tabela MARINESCU\_POMPIERI, evidențiindu-se pentru fiecare pompier nivelul, și superiorii săi.*

* SELECT

LPAD(' ', 4 \* (LEVEL - 1)) || ID\_POMPIER AS M\_ID\_POMPIER, // afișarea ierarhiei în mod indentat

NUME,

PRENUME,

LEVEL AS NIVEL,

SYS\_CONNECT\_BY\_PATH(ID\_POMPIER, ' -> ') AS DRUM

FROM

MARINESCU\_POMPIERI

START WITH

ID\_SUPERIOR IS NULL // începem cu rădăcina arborelui

CONNECT BY

PRIOR ID\_POMPIER = ID\_SUPERIOR // relația părinte-copil

ORDER SIBLINGS BY

ID\_POMPIER;

REZULTAT

