Universitatea Babeș-Bolyai

Facultatea de Științe Economice și Gestiunea Afacerilor

|  |  |
| --- | --- |
| PONTAJ ANGAJAȚI | NECHITA CARINA-ANAMARIA  Specializarea: Informatică Economică  An 3  Grupa 1 |

2020-2021

# Introducere

Baza de date proiectată are în prim plan pontajul angajaților din cadrul unui departament de IT unde se realizează două tipuri de proiecte: aplicații mobile și pagini web, fiecare angajat având diferite sarcini în cadrul acestora.

Există câteva constrângeri cu privire la angajați: angajații care se ocupă de implementarea paginilor web nu se ocupă și de aplicațiile mobile. Pentru angajații responsabili de crearea aplicațiilor mobile sunt necesare studii superioare tehnice finalizate iar pentru dezvoltatorii paginilor web este nevoie de experiență de minim 2 ani în domeniu.

Pontajul este realizat pe lună și va cuprinde și orele suplimentare, stipulate printr-un tarif orar iar în cazul în care un angajat nu se poate prezenta la locul de muncă într-o anumită perioadă de timp din diferite motive, acesta va fi trecut absent.

# Crearea tabelelor

Tabela **JOBURI**

CREATE TABLE joburi(id\_job NUMBER(4) PRIMARY KEY ,denumire VARCHAR2(30) NOT NULL);

ALTER TABLE joburi ADD fisa\_postului CLOB;

Tabela **ANGAJATI**

CREATE TABLE angajati(id\_angajat NUMBER(4) PRIMARY KEY, nume VARCHAR2(25) NOT NULL, prenume VARCHAR2(25) NOT NULL, id\_job NUMBER(4) NOT NULL, email VARCHAR2(20) NOT NULL, data\_nasterii DATE NOT NULL, data\_angajarii DATE NOT NULL, tip\_angajat CHAR(1) NOT NULL, facultate VARCHAR2(10), experienta NUMBER(2));

ALTER TABLE angajati ADD FOREIGN KEY(id\_job) REFERENCES joburi(id\_job);

Tabela **PROGRAM\_LUCRU**

CREATE TABLE program\_lucru(id\_angajat NUMBER(4), data DATE, ora\_start NUMBER(2) NOT NULL, ora\_final NUMBER(2) NOT NULL, PRIMARY KEY(id\_angajat,data));

Tabela **ABSENTE**

CREATE TABLE absente(id\_angajat NUMBER(4), data\_inceput DATE NOT NULL, data\_final DATE NOT NULL, motiv VARCHAR2(15) NOT NULL, PRIMARY KEY(id\_angajat, data\_inceput));

Tabela **PONTAJE**

CREATE TABLE pontaje(id\_angajat NUMBER(4), cod\_data NUMBER(4), ore\_lucrate NUMBER(3) NOT NULL, ore\_suplimentare NUMBER(3), tarif\_orar NUMBER(3), salariu NUMBER(4) NOT NULL,PRIMARY KEY (id\_angajat,cod\_data));

Tabela **CALENDAR**

CREATE TABLE calendar(cod NUMBER(4) PRIMARY KEY, luna VARCHAR2(20) NOT NULL, an NUMBER(4) NOT NULL);

Tabela **PROIECTE**

CREATE TABLE proiecte(id\_proiect NUMBER(4) PRIMARY KEY, denumire VARCHAR2(30) NOT NULL, data\_inceput DATE NOT NULL, data\_final DATE, deadline DATE NOT NULL, status\_proiect VARCHAR2(15) NOT NULL, tip\_proiect CHAR(4) NOT NULL, sistem\_operare VARCHAR2(15));

Tabela **SARCINI**

CREATE TABLE sarcini(id\_sarcina VARCHAR2(4) PRIMARY KEY, denumire VARCHAR2(25) NOT NULL, status\_sarcina VARCHAR2(15) NOT NULL, id\_angajat NUMBER(4) NOT NULL, id\_proiect NUMBER(4) NOT NULL);

ALTER TABLE sarcini ADD FOREIGN KEY(id\_angajat) REFERENCES angajati(id\_angajat);

ALTER TABLE sarcini ADD FOREIGN KEY(id\_proiect) REFERENCES proiecte(id\_proiect);

Tabela **MANAGERI**

CREATE TABLE manageri(id\_angajat NUMBER(4), id\_proiect NUMBER(4), PRIMARY KEY(id\_angajat,id\_proiect));

# Popularea tabelelor

Tabela **JOBURI**

INSERT INTO joburi VALUES (1111,'UI/UX Designer', 'UX-UI Designers are generally responsible for collecting, researching, investigating and evaluating user requirements. Their responsibility is to deliver an outstanding user experience providing an exceptional and intuitive application design.');

INSERT INTO joburi VALUES (2222, 'Back-End Developer', 'Back-end developers create, code, and improve the server, server-side applications, and databases that, when combined with front-end codes, help create a functional, seamless experience for the end-user.');

INSERT INTO joburi VALUES (3333, 'QA Engineer', 'A QA engineer is responsible for the creation of tests to identify issues with software before the product launch.');

INSERT INTO joburi VALUES (4444, 'iOS Developer', 'iOS developers design and build applications for mobile devices running Apples iOS operating software. They are responsible for designing and coding the base application, ensuring the quality of the application, fixing application bugs, maintaining the code, and implementing application updates.');

INSERT INTO joburi VALUES (5555, 'Android Developer', 'An Android developer is responsible for developing applications for devices powered by the Android operating system.');

INSERT INTO joburi VALUES (6666, 'Front-End Developer', 'Front end developers are computer programmers who specialize in website design. Front end developer duties include determining the structure and design of web pages, striking a balance between functional and aesthetic design, and ensuring web design is optimized for smartphones.');

INSERT INTO joburi VALUES (7777, 'Project Manager', 'Project managers are responsible for planning and overseeing projects to ensure they are completed in a timely fashion and within budget. Project managers plan and designate project resources, prepare budgets, monitor progress, and keep stakeholders informed the entire way.');

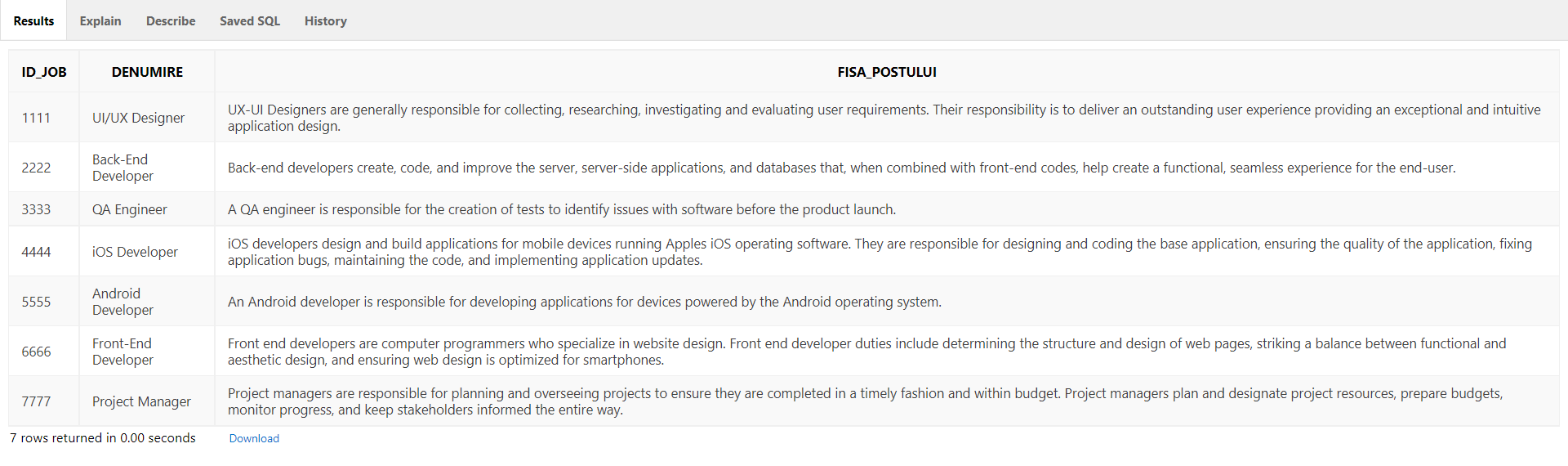


Tabela **ANGAJATI**

INSERT INTO angajati VALUES(1,'Baciu','Valentina',1111,'baciuv@email.ro','28-mar-1996','15-apr-2021','M','UTCN',NULL);

INSERT INTO angajati VALUES(2,'Andronic','Petre',2222,'andronicp@email.ro','15-sep-1985','30-june-2011','W',NULL,3);

INSERT INTO angajati VALUES(3,'Nica','Bianca',1111,'nicab@email.ro','29-oct-1997','10-jan-2021','W',NULL,4);

INSERT INTO angajati VALUES(4,'Deac','Clara',2222,'deacc@email.ro','05-jan-1980','05-aug-2015','W',NULL,2);

INSERT INTO angajati VALUES(5,'Sabou','Aurel',2222,'saboua@email.ro','12-dec-1995','17-may-2020','M','FSEGA',NULL);

INSERT INTO angajati VALUES(6,'Alb','David',3333,'albd@email.ro','24-mar-1988','02-sep-2010','M','MI',NULL);

INSERT INTO angajati VALUES(7,'Dancea','Emil',6666,'danceae@email.ro','13-june-1972','24-nov-2011','W',NULL,5);

INSERT INTO angajati VALUES(8,'Nuc','George',4444,'nucg@email.ro','09-aug-1992','19-dec-2017','M','UTCN',NULL);

INSERT INTO angajati VALUES(9,'Anton','Adina',5555,'antona@email.ro','16-dec-1999','21-aug-2021','M','FSEGA',NULL);

INSERT INTO angajati VALUES(10,'Balan','Dragos',3333,'baland@email.ro','31-july-1980','22-oct-2013','W',NULL,2);

INSERT INTO angajati VALUES(11,'Cepu','Antonia',7777,'cepua@email.ro','21-apr-1983','03-june-2014','M','MI',NULL);

INSERT INTO angajati VALUES(12,'Rica','Nicoleta',7777,'rican@email.ro','11-oct-1984','19-nov-2016','W',NULL,5);

INSERT INTO angajati VALUES(13,'Hidan','Cornel',7777,'hidanc@email.ro','29-nov-1973','18-oct-2013','M','MI',NULL);

INSERT INTO angajati VALUES(14,'Dicu','Liviu',7777,'dicul@email.ro','06-may-1977','25-mar-2021','W',NULL,6);



Tabela **PROIECTE**

INSERT INTO proiecte VALUES(1234,'Smart Parking','01-feb-2021','15-apr-2021','15-apr-2021','finalizat','AM','android');

INSERT INTO proiecte VALUES(1235,'Online Store','05-dec-2021','21-dec-2021','17-dec-2021','intarziat','AW',NULL);

INSERT INTO proiecte VALUES(1236,'Smart City Traveler','13-jan-2021',NULL,'28-aug-2021','nefinalizat','AM','iOS');

INSERT INTO proiecte VALUES(1237,'Voting System','20-mar-2021',NULL,'30-jan-2022','nefinalizat','AM','android');

INSERT INTO proiecte VALUES(1238,'Business Website','19-mar-2021',NULL,'03-sep-2021','nefinalizat','AW',NULL);

INSERT INTO proiecte VALUES(1239,'Hotel Reservation','07-oct-2021','21-oct-2021','22-oct-2021','finalizat','AM','android');

INSERT INTO proiecte VALUES(1244,'Car Website','18-oct-2021',NULL,'15-dec-2021','intarziat','AW',NULL);

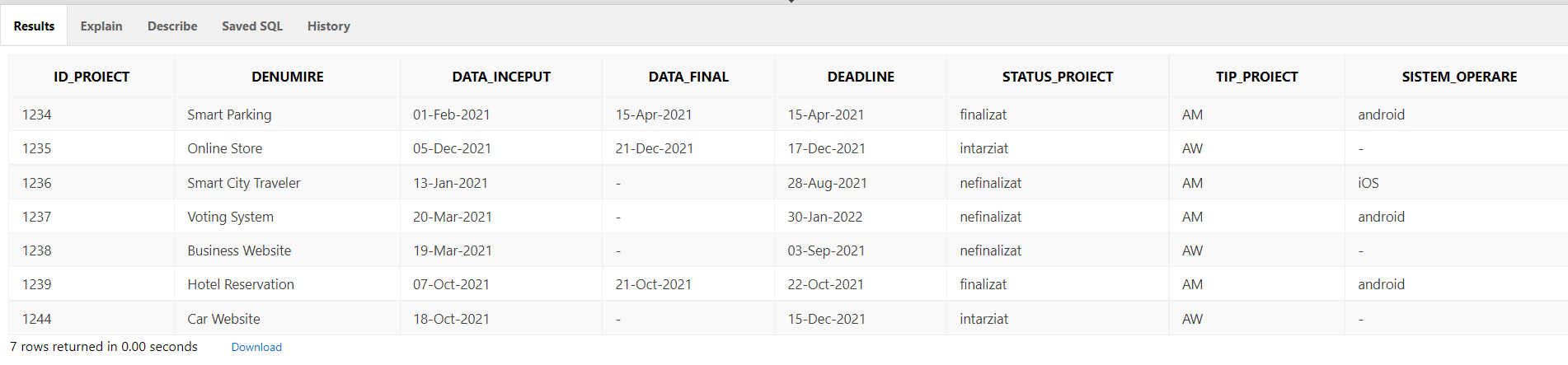


Tabela **SARCINI**

INSERT INTO sarcini VALUES('sm1','schita aplicatie','finalizata',1,1234);

INSERT INTO sarcini VALUES('sm2','interfata aplicatie','finalizata',1,1234);

INSERT INTO sarcini VALUES('sm3','cod','finalizata',9,1234);

INSERT INTO sarcini VALUES('sm4','arhitectura aplicatie','finalizata',4,1234);

INSERT INTO sarcini VALUES('sm5','testare','finalizata',6,1234);

INSERT INTO sarcini VALUES('sw6','wireframe','finalizata',3,1238);

INSERT INTO sarcini VALUES('sw7','bugs','nefinalizata',10,1238);

INSERT INTO sarcini VALUES('sw8','cod','finalizata',7,1238);

INSERT INTO sarcini VALUES('sw9','cod','nefinalizata',4,1238);

INSERT INTO sarcini VALUES('mm1','planificare','finalizata',11,1234);

INSERT INTO sarcini VALUES('mm2','implementare backup','finalizata',11,1234);

INSERT INTO sarcini VALUES('mm3','semnare contract','finalizata',12,1238);

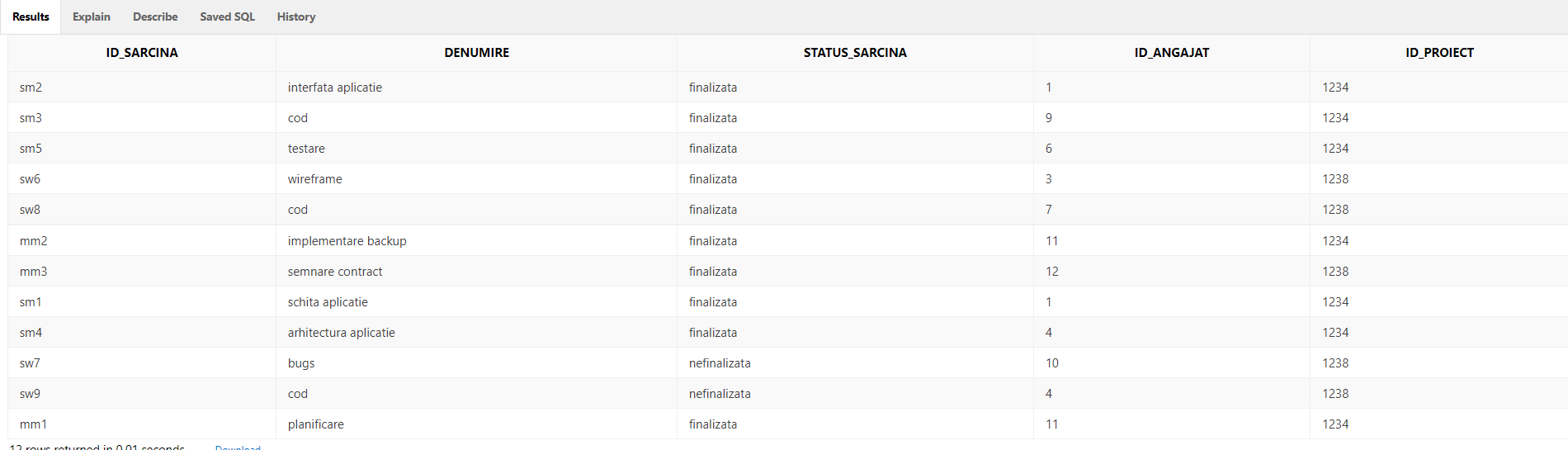


Tabela **PONTAJE**

INSERT INTO pontaje VALUES(1,1221,50,3 ,10,4000);

INSERT INTO pontaje VALUES(2,1221,60,0,NULL,5500);

INSERT INTO pontaje VALUES(3,1221,55,0,NULL,6000);

INSERT INTO pontaje VALUES(4,1221,45,0,NULL,3900);

INSERT INTO pontaje VALUES(5,1221,45,5,15,3900);

INSERT INTO pontaje VALUES(6,1221,50,4,12,4100);

INSERT INTO pontaje VALUES(7,1221,32,1,9,3000);

INSERT INTO pontaje VALUES(8,1221,48,0,NULL,3950);

INSERT INTO pontaje VALUES(9,1221,63,0,NULL,5500);

INSERT INTO pontaje VALUES(10,1221,58,0,NULL,4780);

INSERT INTO pontaje VALUES(11,1221,48,0,NULL,6500);

INSERT INTO pontaje VALUES(12,1221,60,0,NULL,6550);

INSERT INTO pontaje VALUES(13,1221,52,0,NULL,6700);

INSERT INTO pontaje VALUES(14,1221,44,0,NULL,6890);

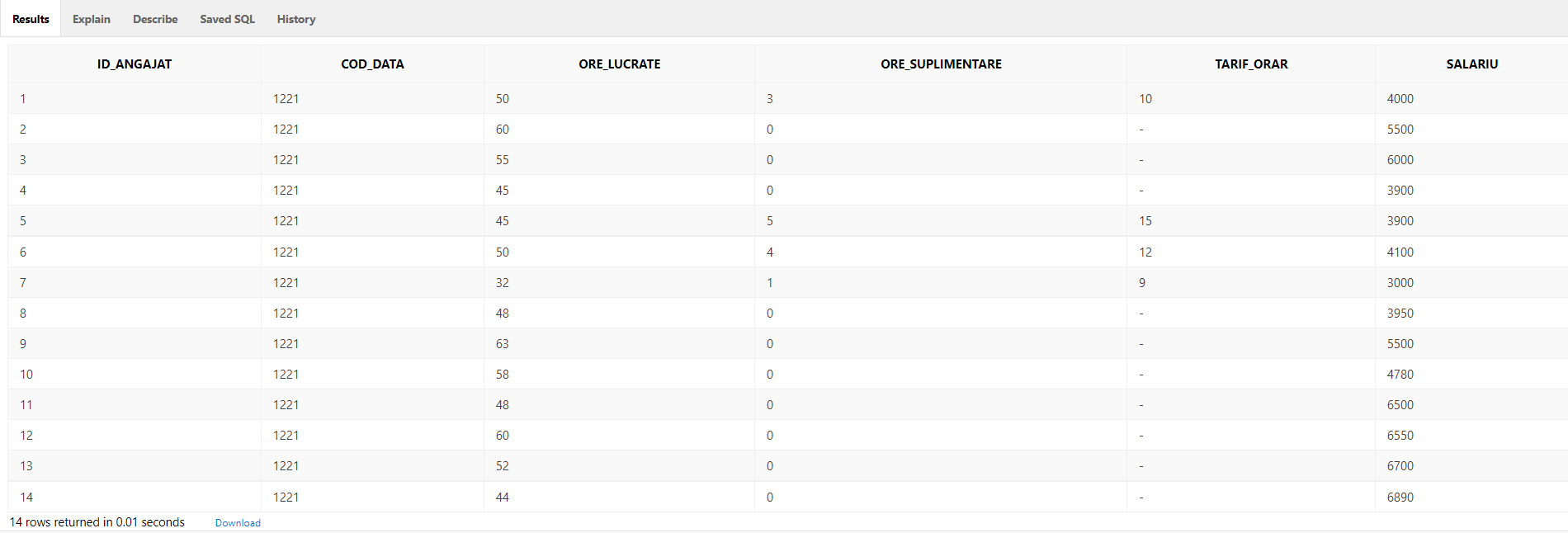


Tabela **CALENDAR**

INSERT INTO calendar VALUES(0121,'ianuarie',2021);

INSERT INTO calendar VALUES(0221,'februarie',2021);

INSERT INTO calendar VALUES(0321,'martie',2021);

INSERT INTO calendar VALUES(0421,'aprilie',2021);

INSERT INTO calendar VALUES(0521,'mai',2021);

INSERT INTO calendar VALUES(0621,'iunie',2021);

INSERT INTO calendar VALUES(0721,'iulie',2021);

INSERT INTO calendar VALUES(0821,'august',2021);

INSERT INTO calendar VALUES(0921,'septembrie',2021);

INSERT INTO calendar VALUES(1021,'octombrie',2021);

INSERT INTO calendar VALUES(1121,'noiembrie',2021);

INSERT INTO calendar VALUES(1221,'decembrie',2021);

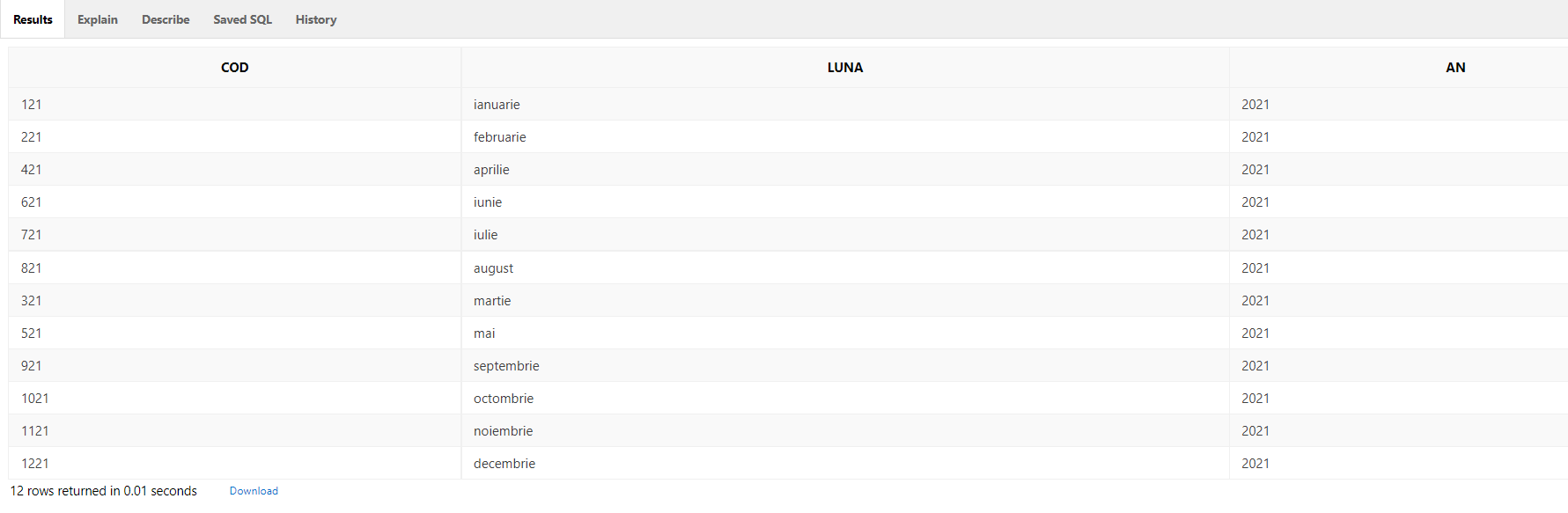


Tabela **ABSENTE**

INSERT INTO absente VALUES(2,'05-dec-2021','06-dec-2021','medical');

INSERT INTO absente VALUES(7,'13-dec-2021','20-dec-2021','concediu');

INSERT INTO absente VALUES(2,'12-dec-2021','14-dec-2021','necunoscut');

INSERT INTO absente VALUES(8,'12-dec-2021','15-dec-2021','medical');

INSERT INTO absente VALUES(7,'20-dec-2021','22-dec-2021','medical');

INSERT INTO absente VALUES(6,'20-dec-2021','22-dec-2021','concediu');

INSERT INTO absente VALUES(12,'21-dec-2021','22-dec-2021','medical');

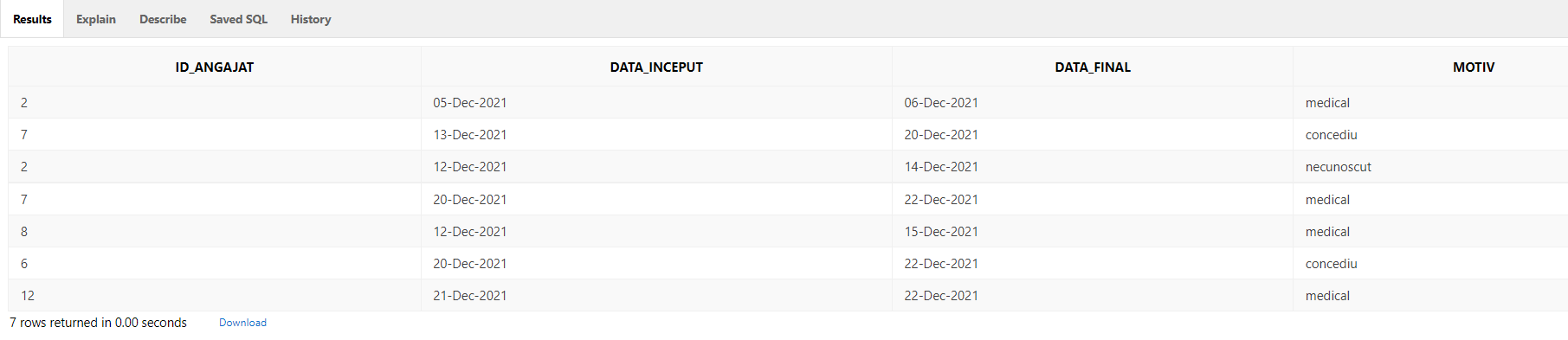


Tabela **PROGRAM\_LUCRU**

INSERT INTO program\_lucru VALUES(1,'05-dec-2021',8,13);

INSERT INTO program\_lucru VALUES(2,'05-dec-2021',8,16);

INSERT INTO program\_lucru VALUES(3,'05-dec-2021',9,13);

INSERT INTO program\_lucru VALUES(4,'05-dec-2021',10,17);

INSERT INTO program\_lucru VALUES(5,'05-dec-2021',8,15);

INSERT INTO program\_lucru VALUES(6,'05-dec-2021',13,19);

INSERT INTO program\_lucru VALUES(7,'05-dec-2021',9,16);

INSERT INTO program\_lucru VALUES(8,'05-dec-2021',8,16);

INSERT INTO program\_lucru VALUES(9,'05-dec-2021',8,18);

INSERT INTO program\_lucru VALUES(10,'05-dec-2021',9,18);

INSERT INTO program\_lucru VALUES(11,'05-dec-2021',8,15);

INSERT INTO program\_lucru VALUES(12,'05-dec-2021',9,16);

INSERT INTO program\_lucru VALUES(13,'05-dec-2021',8,17);

INSERT INTO program\_lucru VALUES(14,'05-dec-2021',8,18);

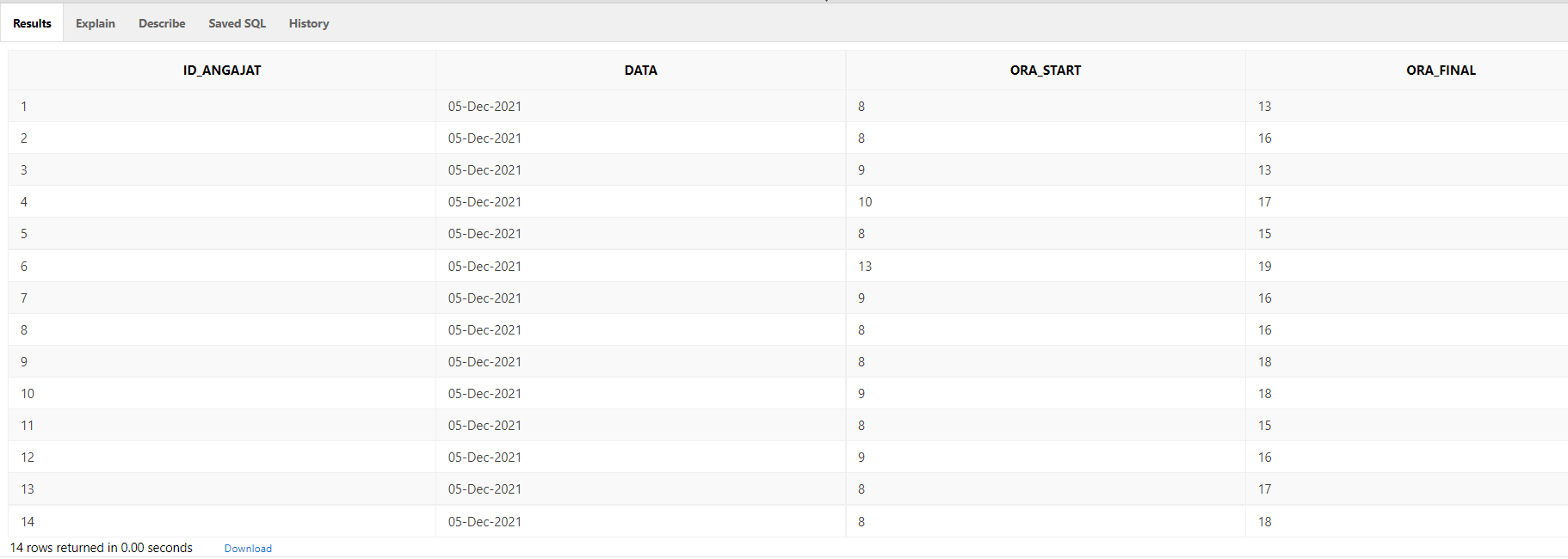


Tabela **MANAGERI**

INSERT INTO manageri VALUES(11,1234);

INSERT INTO manageri VALUES(12,1235);

INSERT INTO manageri VALUES(13,1236);

INSERT INTO manageri VALUES(11,1237);

INSERT INTO manageri VALUES(12,1238);

INSERT INTO manageri VALUES(11,1239);

INSERT INTO manageri VALUES(12,1244);

# Modificări de structură

1. Tabela Pontaje- se adaugă coloana “penalizări” cu valoarea implicită 0

ALTER TABLE pontaje ADD penalizari NUMBER(5) DEFAULT 0;

1. Tabela Pontaje- adăugarea check-ului pentru salariu (salariu>0)

ALTER TABLE pontaje ADD CONSTRAINT check\_salariu CHECK(salariu>0);

1. Tabela Angajati- adăugarea check-ului pentru experiență (experienta>=2)

ALTER TABLE angajati ADD CONSTRAINT check\_experienta CHECK(experienta>=2);

# Package

create or replace PACKAGE pack\_proiect AS

procedure adaugare\_proiect(p\_id\_proiect proiecte.id\_proiect%TYPE, p\_denumire proiecte.denumire%TYPE, p\_data\_inceput proiecte.data\_inceput%TYPE, p\_data\_final proiecte.data\_final%TYPE, p\_deadline proiecte.deadline%TYPE, p\_status proiecte.status\_proiect%TYPE, p\_tip proiecte.tip\_proiect%TYPE, p\_sistem\_operare proiecte.sistem\_operare%TYPE);

procedure marire\_salariu\_Craciun;

procedure absente\_angajati(p\_motiv absente.motiv%TYPE);

function vechime\_maxima\_angajat RETURN NUMBER;

procedure lista\_angajati(p\_cod\_data pontaje.cod\_data%TYPE);

END;

create or replace PACKAGE BODY pack\_proiect AS

SO1 CONSTANT VARCHAR2(10):= 'android';

SO2 CONSTANT VARCHAR2(10):= 'iOS';

eSOIncorect exception;

procedure msg\_confirmare IS

BEGIN

IF SQL%ROWCOUNT>0 THEN

DBMS\_OUTPUT.PUT\_LINE('Proiect adaugat cu succes');

END IF;

END;

procedure adaugare\_proiect(p\_id\_proiect proiecte.id\_proiect%TYPE, p\_denumire proiecte.denumire%TYPE, p\_data\_inceput proiecte.data\_inceput%TYPE, p\_data\_final proiecte.data\_final%TYPE, p\_deadline proiecte.deadline%TYPE, p\_status proiecte.status\_proiect%TYPE, p\_tip proiecte.tip\_proiect%TYPE, p\_sistem\_operare proiecte.sistem\_operare%TYPE) IS

BEGIN

IF p\_sistem\_operare != SO1 OR p\_sistem\_operare != SO2 THEN

RAISE eSOIncorect;

END IF;

INSERT INTO proiecte VALUES(p\_id\_proiect, p\_denumire, p\_data\_inceput, p\_data\_final, p\_deadline, p\_status, p\_tip, p\_sistem\_operare);

msg\_confirmare();

EXCEPTION

WHEN eSOIncorect THEN

DBMS\_OUTPUT.PUT\_LINE('sistem operare invalid');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE('Exista deja un proiect cu id-ul ' || p\_id\_proiect);

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('EROARE');

END;

procedure bonus(p\_id\_angajat IN pontaje.id\_angajat%TYPE, p\_cod\_data IN pontaje.cod\_data%TYPE, p\_percent IN NUMBER) IS

BEGIN

UPDATE pontaje

SET salariu = salariu \* (1 + p\_percent/100)

WHERE id\_angajat = p\_id\_angajat AND cod\_data = p\_cod\_data;

END;

procedure marire\_salariu\_Craciun IS

var\_ang pontaje.id\_angajat%TYPE;

CURSOR ang\_crs IS SELECT id\_angajat FROM pontaje;

BEGIN

OPEN ang\_crs;

LOOP

FETCH ang\_crs INTO var\_ang;

EXIT WHEN ang\_crs%NOTFOUND;

bonus(var\_ang,1221,10);

END LOOP;

CLOSE ang\_crs;

END;

procedure absente\_angajati(p\_motiv IN absente.motiv%TYPE) IS

CURSOR angajat\_crs (p\_id\_angajat NUMBER) IS SELECT id\_angajat, nume, prenume FROM angajati WHERE id\_angajat = p\_id\_angajat;

CURSOR motiv\_crs IS SELECT id\_angajat, motiv FROM absente WHERE p\_motiv = motiv;

var\_motiv motiv\_crs%ROWTYPE;

var\_ang angajat\_crs%ROWTYPE;

BEGIN

OPEN motiv\_crs;

LOOP

FETCH motiv\_crs INTO var\_motiv;

EXIT WHEN motiv\_crs%NOTFOUND;

OPEN angajat\_crs(var\_motiv.id\_angajat);

LOOP

FETCH angajat\_crs INTO var\_ang;

EXIT WHEN angajat\_crs%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(var\_ang.nume|| ' ' || var\_ang.prenume);

END LOOP;

CLOSE angajat\_crs;

END LOOP;

CLOSE motiv\_crs;

END;

function vechime\_maxima\_angajat RETURN NUMBER IS

var\_id\_ang angajati.id\_angajat%TYPE;

var\_data\_vechime angajati.data\_angajarii%TYPE;

BEGIN

SELECT MIN(data\_angajarii)

INTO var\_data\_vechime

FROM angajati;

SELECT id\_angajat

INTO var\_id\_ang

FROM angajati

WHERE data\_angajarii = var\_data\_vechime;

RETURN var\_id\_ang;

END;

function ore\_lucrate\_medie(f\_cod\_data IN pontaje.cod\_data%TYPE) return number is

var\_ore\_med number;

BEGIN

SELECT AVG(ore\_lucrate)

INTO var\_ore\_med

FROM pontaje

WHERE f\_cod\_data = cod\_data;

RETURN var\_ore\_med;

END;

procedure lista\_angajati(p\_cod\_data IN pontaje.cod\_data%TYPE) IS

CURSOR lista\_ang\_crs(p\_ore\_lucrate NUMBER) IS SELECT id\_angajat FROM pontaje WHERE ore\_lucrate>p\_ore\_lucrate;

var\_ore NUMBER;

var\_lista\_ang lista\_ang\_crs%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Angajatii care au lucrat peste media orelor');

var\_ore := ore\_lucrate\_medie(1221);

OPEN lista\_ang\_crs(var\_ore);

LOOP

FETCH lista\_ang\_crs INTO var\_lista\_ang;

EXIT WHEN lista\_ang\_crs%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(var\_lista\_ang.id\_angajat);

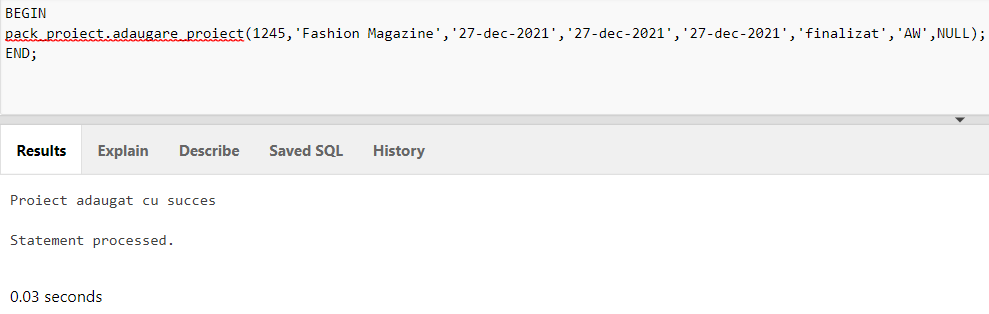
END LOOP;

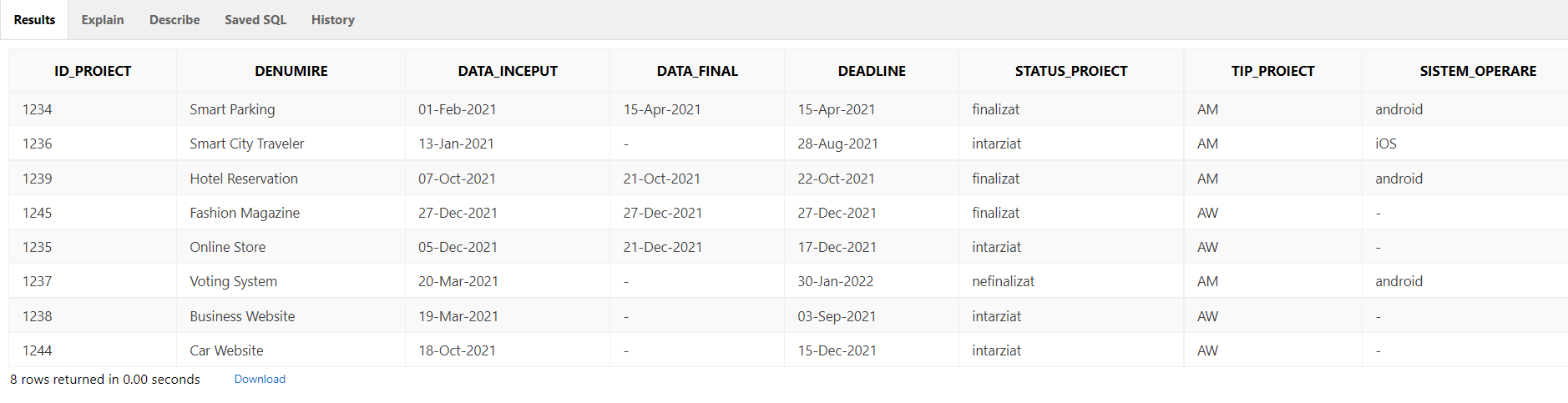
END;

END;

1. procedura privată „msg\_confirmare”

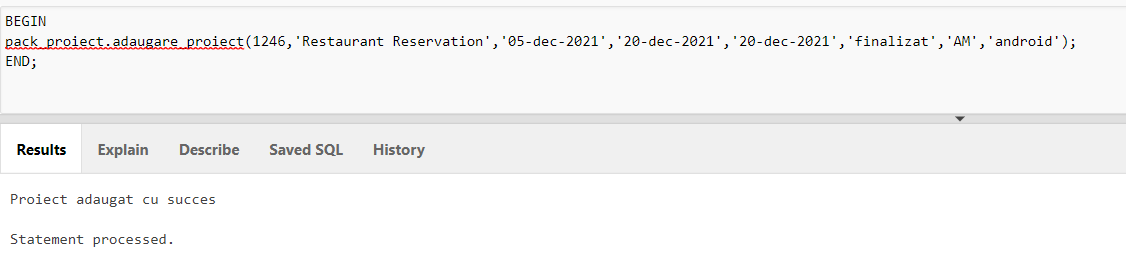
Se utilizează în cadrul procedurii publice „adaugare\_proiect” pentru a afișa un mesaj de succes.



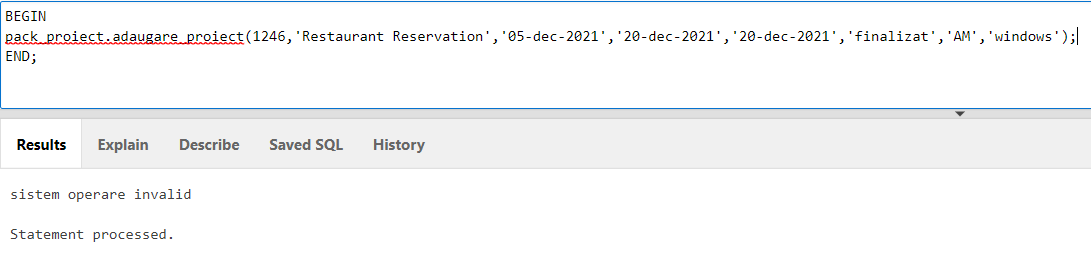


1. procedura publică “adaugare\_proiect”

Se utilizează pentru adăugarea unui nou proiect în tabela “proiecte”.







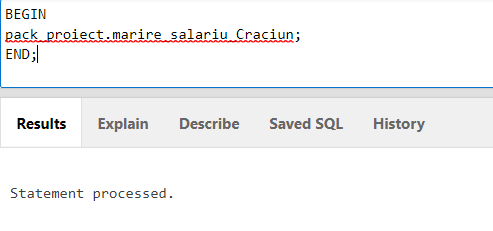


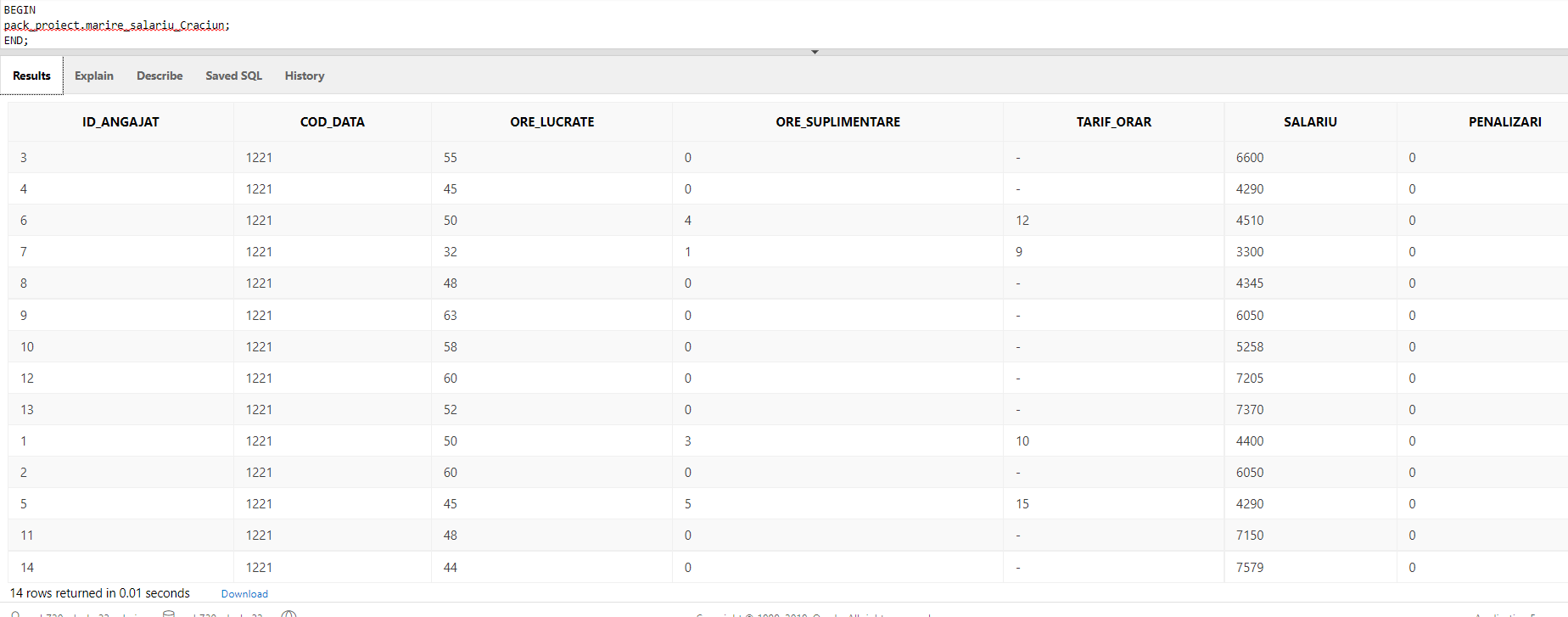
1. procedura privată “bonus”

Calculează noul salariu după aplicarea unui bonus specificat pentru o anumită lună și un anumit angajat. Se utilizează în procedura publică “marire\_salariu\_Craciun”.

1. procedura publică “marire\_salariu\_Craciun”

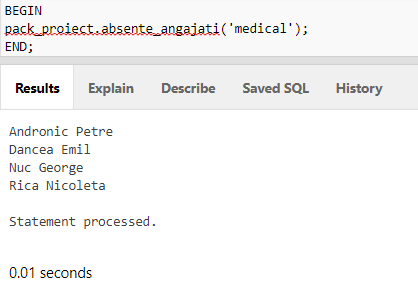
Oferă o mărire de 10% a salariului tuturor angajaților pe luna decembrie.





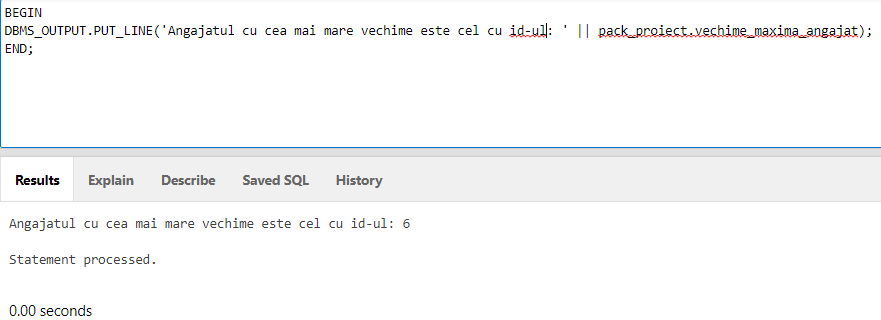
1. procedura publică “absente angajati”

Afișează lista angajaților care au lipsit dintr-un anumit motiv.



1. funcția publică „vechime\_maxima\_angajat”

Afișează id-ul angajatului cu cea mai mare vechime.

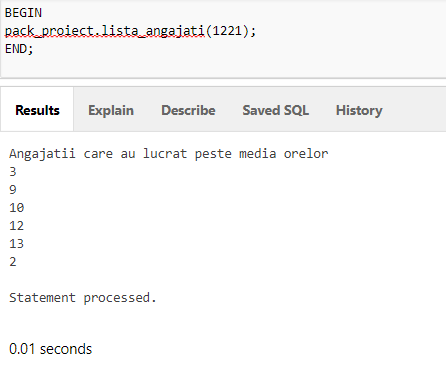


1. funcția privată “ore\_lucrate\_medie”

Calculează media orelor lucrate pe lună. Va fi apelată în procedura publică “lista\_angajati”.

1. procedura publică “lista\_angajati”

Afișează angajații dintr-o anumită lună care au lucrat peste media orelor dintr-o lună.



# Triggeri

1. Actualizează automat statusul proiectului în finalizat după introducerea datei de final a proiectului.

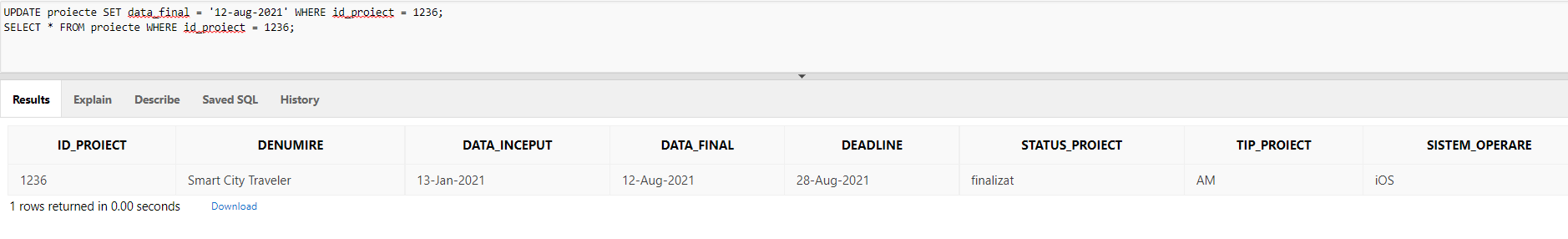
create or replace TRIGGER actualizare\_status\_trigg

AFTER UPDATE OF data\_final ON proiecte

BEGIN

UPDATE proiecte SET status\_proiect = 'finalizat';

END;



1. Afișează vechiul salariu, noul salariu și diferența dintre ele

create or replace TRIGGER afisare\_info\_sal\_trigg

BEFORE UPDATE OF salariu ON pontaje

FOR EACH ROW

DECLARE

diferenta\_sal NUMBER;

BEGIN

diferenta\_sal := :NEW.salariu - :OLD.salariu;

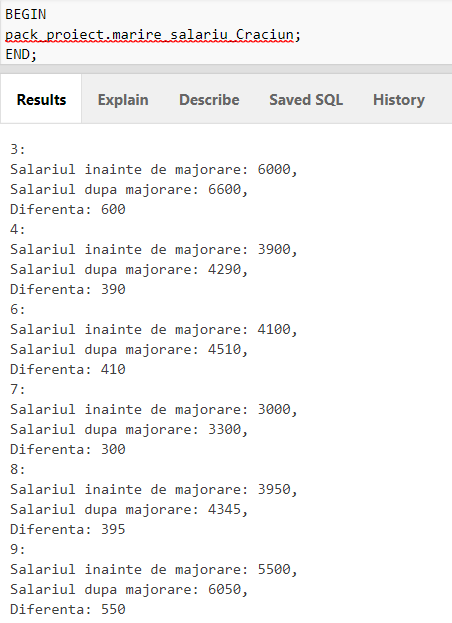
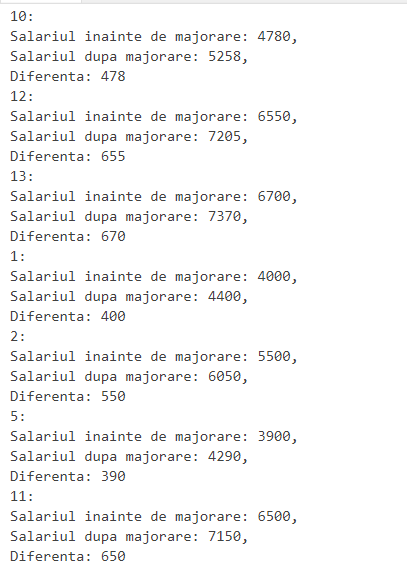
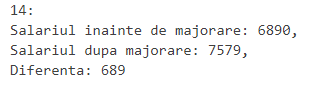
DBMS\_OUTPUT.PUT\_LINE(:NEW.id\_angajat || ': ');

DBMS\_OUTPUT.PUT\_LINE('Salariul inainte de majorare: ' || :OLD.salariu || ', ');

DBMS\_OUTPUT.PUT\_LINE('Salariul dupa majorare: ' || :NEW.salariu || ', ');

DBMS\_OUTPUT.PUT\_LINE('Diferenta: ' || diferenta\_sal);

END;



1. Afișează mesaje specifice dacă se dorește inserarea/ actualizarea asupra tabelei program\_lucru

create or replace TRIGGER program\_trigg

BEFORE INSERT OR UPDATE ON program\_lucru

BEGIN

IF TO\_CHAR(SYSDATE, 'HH24') NOT BETWEEN '08' AND '20' THEN

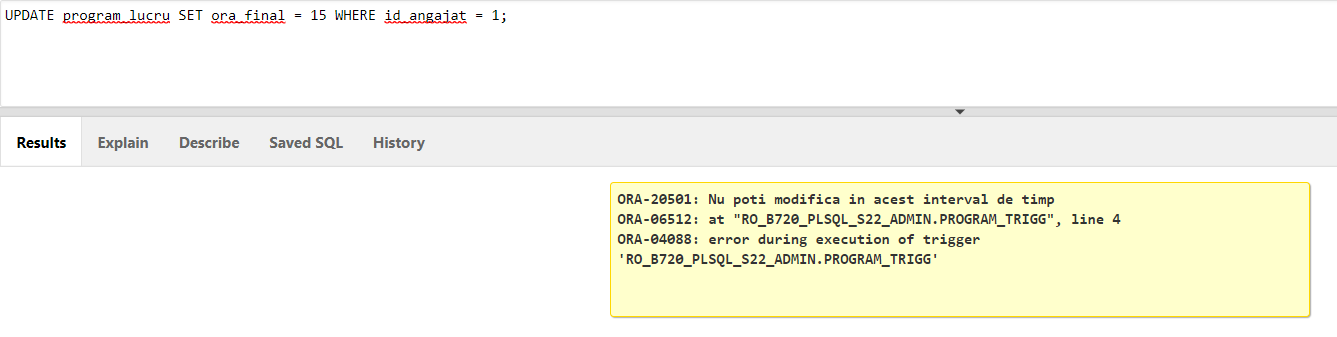
IF INSERTING THEN RAISE\_APPLICATION\_ERROR(-20500, 'Nu poti insera in acest interval de timp');

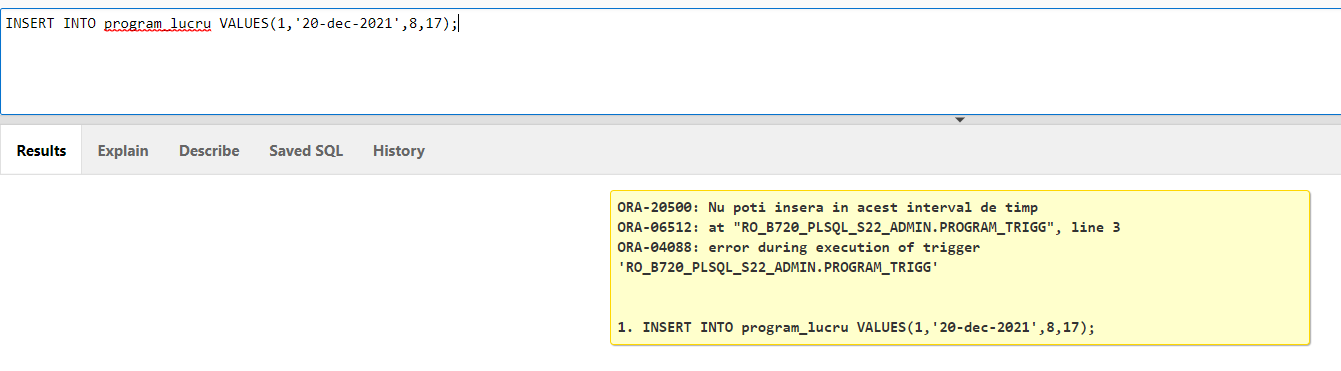
ELSE RAISE\_APPLICATION\_ERROR(-20501, 'Nu poti modifica in acest interval de timp');

END IF;

END IF;

END;





1. Afișează un mesaj de eroare dacă se dorește inserarea unui angajat cu vârsta sub 18 ani.

create or replace TRIGGER verif\_varsta

BEFORE INSERT ON angajati

FOR EACH ROW

DECLARE

varsta\_ang NUMBER;

BEGIN

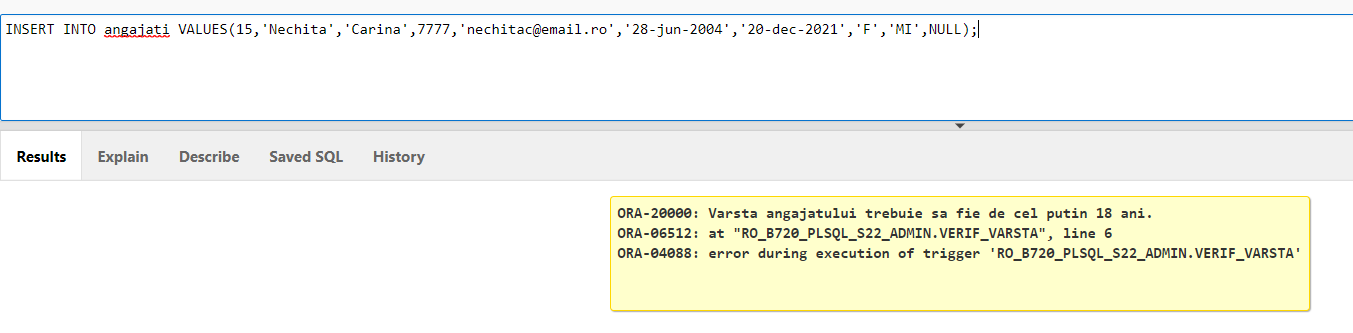
SELECT MONTHS\_BETWEEN(TO\_DATE(sysdate,'DD-MON-YYYY'), TO\_DATE(:new.data\_nasterii,'DD-MON-YYYY'))/12 INTO varsta\_ang FROM DUAL;

IF varsta\_ang < 18 THEN

RAISE\_APPLICATION\_ERROR(-20000,'Varsta angajatului trebuie sa fie de cel putin 18 ani.');

END IF;

END;



# Secvențe

1. create SEQUENCE angajat\_seq

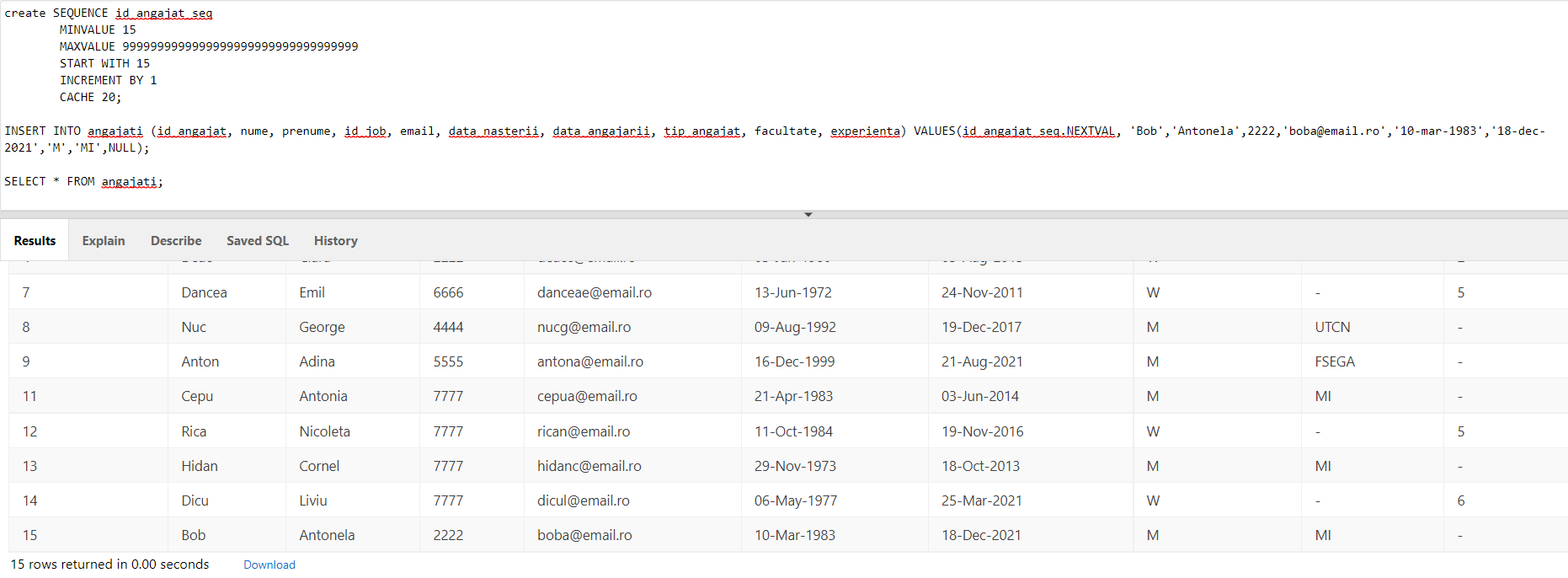
MINVALUE 15

MAXVALUE 9999999999999999999999999999999999

START WITH 15

INCREMENT BY 1

CACHE 20;



1. create SEQUENCE proiect\_seq

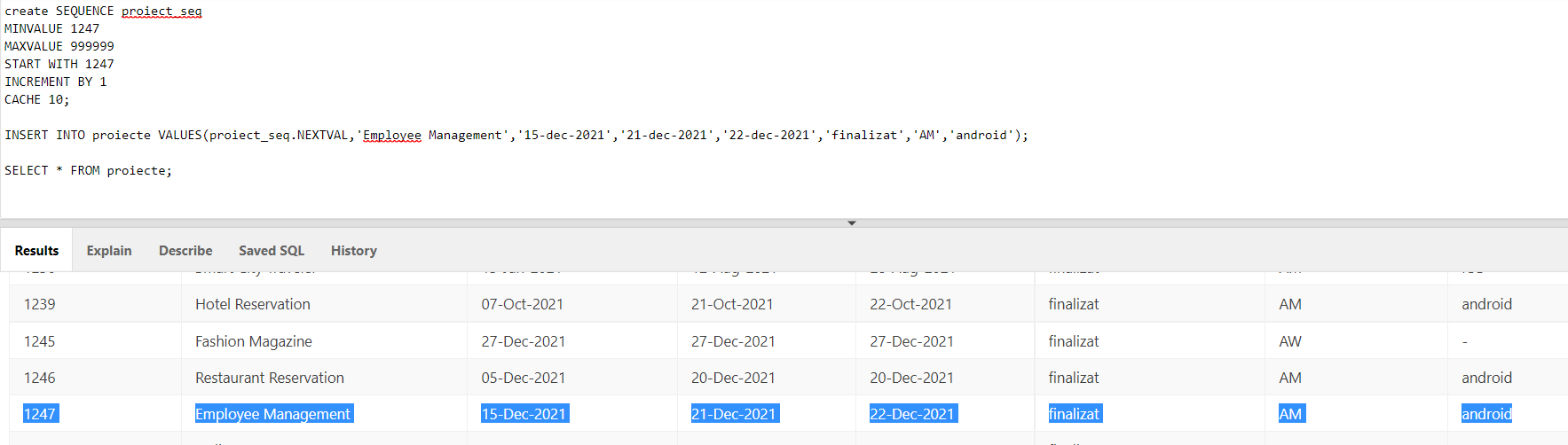
MINVALUE 1247

MAXVALUE 999999

START WITH 1247

INCREMENT BY 1

CACHE 10;



1. create SEQUENCE job\_seq

MINVALUE 8888

MAXVALUE 999999999999999999999999999999999999999

START WITH 8888

INCREMENT BY 1111

CACHE 30;

