GESTIUNEA UNEI APLICATII PENTRU LIVRAREA DE PRODUSE

POPA IULIA-ANDREEA

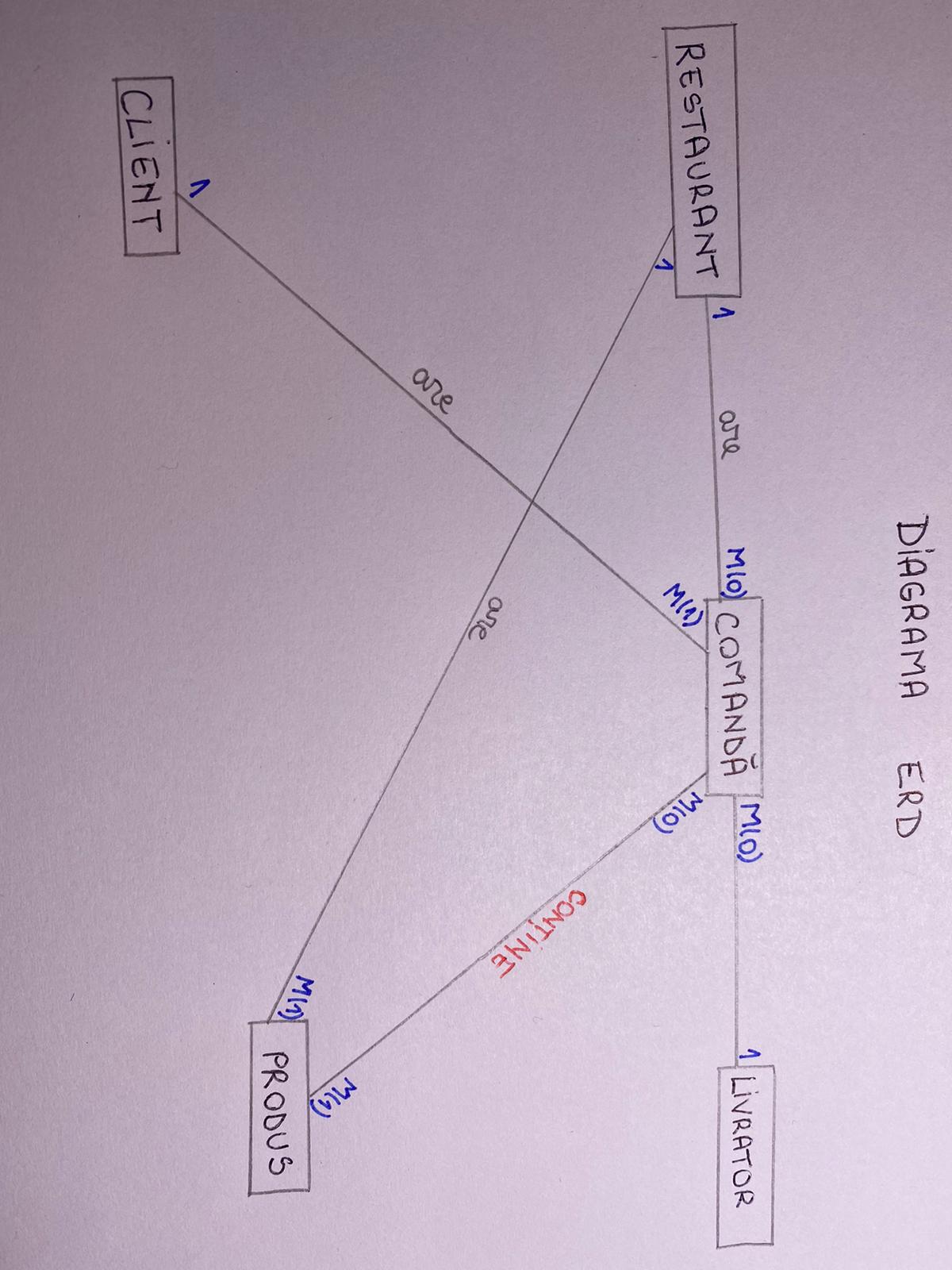
GRUPA 241

PROIECT

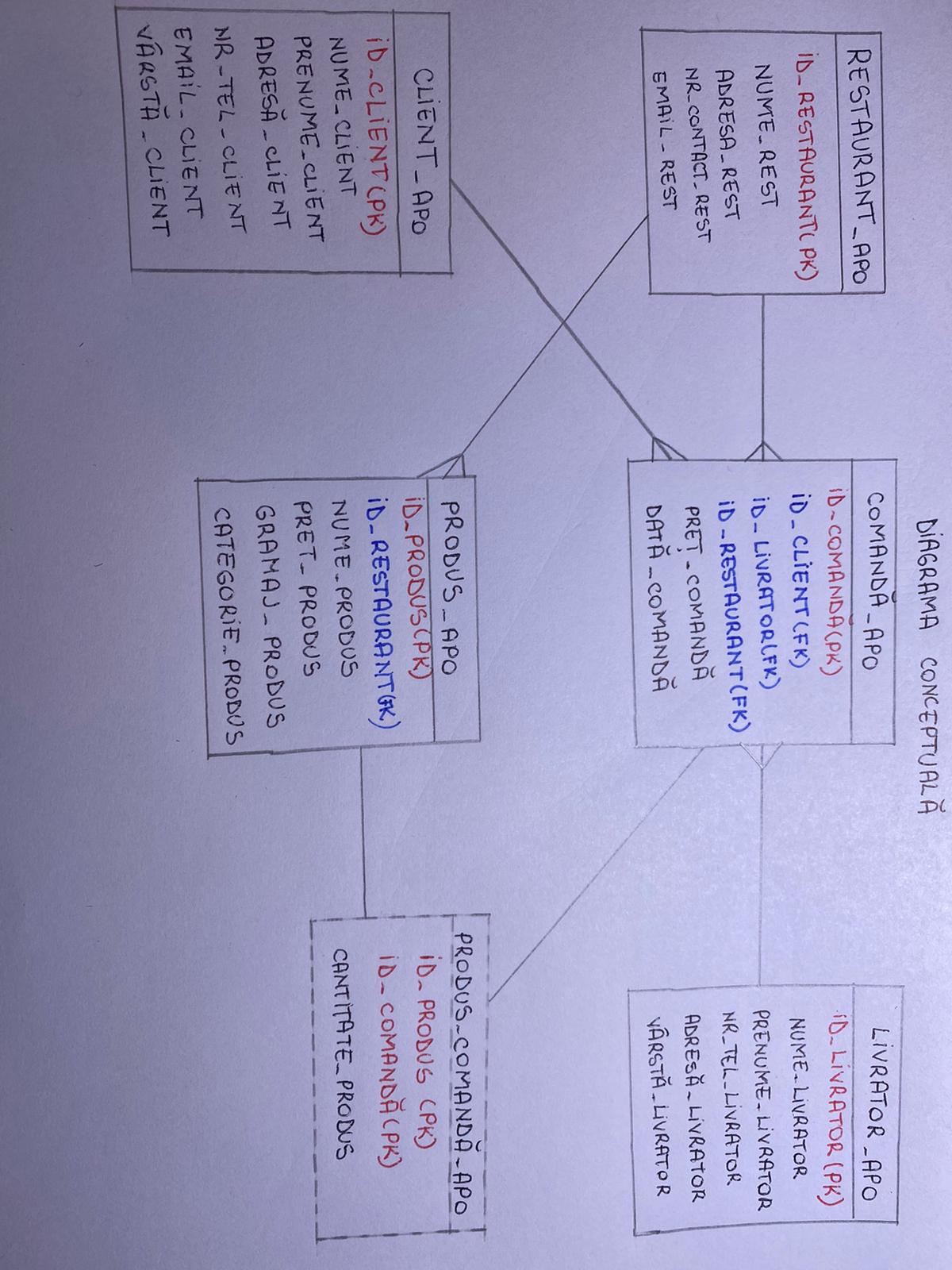
1. Prezentați pe scurt baza de date (utilitatea ei).

In cadrul acestui proiect am creat o baza de date necesara unei aplicatii care se ocupa cu plasarea comenzilor de produse alimentare de la diferite restaurante. Aceasta stocheaza informatii despre fiecare comanda plasata (COMANDA\_APO), restaurantele disponibile (RESTAURANT\_APO) si produsele din meniul acestora (PRODUS\_APO), livratorii angajati (LIVRATOR\_APO), istoricul clientilor (CLIENT\_APO), cat si despre produsele comandate (PRODUS\_COMANDA\_APO) la fiecare comanda.

1. Realizați diagrama entitate-relație (ERD).



1. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



1. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

CREATE TABLE restaurant\_apo(

id\_restaurant NUMBER(4) NOT NULL PRIMARY KEY,

nume\_rest VARCHAR2(30) NOT NULL,

adresa\_rest VARCHAR2(80),

nr\_contact\_rest VARCHAR2(12),

email\_rest VARCHAR2(50));

CREATE TABLE comanda\_apo(

id\_comanda NUMBER(4) NOT NULL PRIMARY KEY,

id\_client NUMBER(4) NOT NULL,

id\_livrator NUMBER(4) NOT NULL,

id\_restaurant NUMBER(4) NOT NULL,

pret\_comanda NUMBER(4) NOT NULL,

data\_comanda DATE NOT NULL);

CREATE TABLE client\_apo(

id\_client NUMBER(4) NOT NULL PRIMARY KEY,

nume\_client VARCHAR2(20) NOT NULL,

prenume\_client VARCHAR2(20) NOT NULL,

adresa\_client VARCHAR2(80) NOT NULL,

nr\_tel\_client VARCHAR2(12) NOT NULL,

email\_client VARCHAR2(50),

varsta\_client NUMBER(4));

CREATE TABLE produs\_apo(

id\_produs NUMBER(4) NOT NULL PRIMARY KEY,

id\_restaurant NUMBER(4) NOT NULL,

nume\_produs VARCHAR2(20) NOT NULL,

pret\_produs NUMBER(4) NOT NULL,

gramaj\_produs NUMBER(4),

categorie\_produs VARCHAR2(30));

CREATE TABLE produs\_comanda\_apo(

id\_produs NUMBER(4) NOT NULL,

id\_comanda NUMBER(4) NOT NULL,

cantitate\_produs NUMBER(4) NOT NULL);

ALTER TABLE produs\_comanda\_apo

ADD CONSTRAINT produs\_comanda\_apo\_PK PRIMARY KEY (id\_produs, id\_comanda);

CREATE TABLE livrator\_apo(

id\_livrator NUMBER(4) NOT NULL PRIMARY KEY,

nume\_livrator VARCHAR2(20) NOT NULL,

prenume\_livrator VARCHAR2(20) NOT NULL,

nr\_tel\_livrator VARCHAR2(12) NOT NULL,

adresa\_livrator VARCHAR2(80),

varsta\_livrator NUMBER(4));

ALTER TABLE comanda\_apo

ADD CONSTRAINT comanda\_apo\_FK1 FOREIGN KEY (id\_client) REFERENCES client\_apo(id\_client);

ALTER TABLE comanda\_apo

ADD CONSTRAINT comanda\_apo\_FK2 FOREIGN KEY (id\_livrator) REFERENCES livrator\_apo(id\_livrator);

ALTER TABLE comanda\_apo

ADD CONSTRAINT comanda\_apo\_FK3 FOREIGN KEY (id\_restaurant) REFERENCES restaurant\_apo(id\_restaurant);

ALTER TABLE produs\_apo

ADD CONSTRAINT produs\_apo\_FK1 FOREIGN KEY (id\_restaurant) REFERENCES restaurant\_apo(id\_restaurant);

1. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

INSERT INTO restaurant\_apo VALUES (1, 'Manor', 'Craiova, Strada Severinului 17', '0756435785', 'restaurant\_manor@gmail.com');

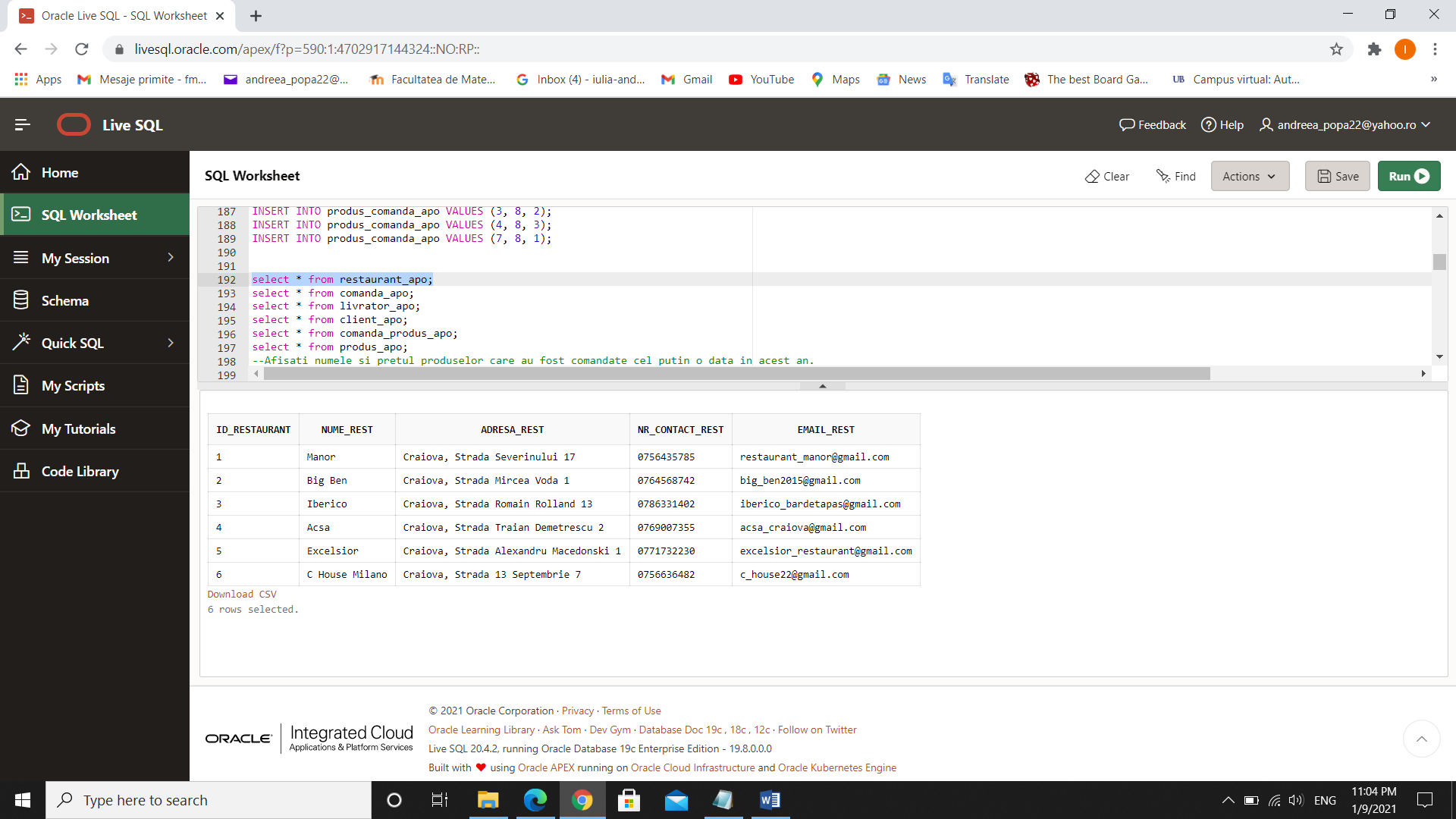
INSERT INTO restaurant\_apo VALUES (2, 'Big Ben', 'Craiova, Strada Mircea Voda 1', '0764568742', 'big\_ben2015@gmail.com');

INSERT INTO restaurant\_apo VALUES (3, 'Iberico', 'Craiova, Strada Romain Rolland 13', '0786331402', 'iberico\_bardetapas@gmail.com');

INSERT INTO restaurant\_apo VALUES (4, 'Acsa', 'Craiova, Strada Traian Demetrescu 2', '0769007355', 'acsa\_craiova@gmail.com');

INSERT INTO restaurant\_apo VALUES (5, 'Excelsior', 'Craiova, Strada Alexandru Macedonski 1', '0771732230', 'excelsior\_restaurant@gmail.com');

INSERT INTO restaurant\_apo VALUES (6, 'C House Milano', 'Craiova, Strada 13 Septembrie 7', '0756636482', 'c\_house22@gmail.com');



INSERT INTO livrator\_apo VALUES (1, 'Stefanescu', 'Codrin Mihai', '0765458630', 'Craiova, Str Severin 19, bl M19, ap 3', '23');

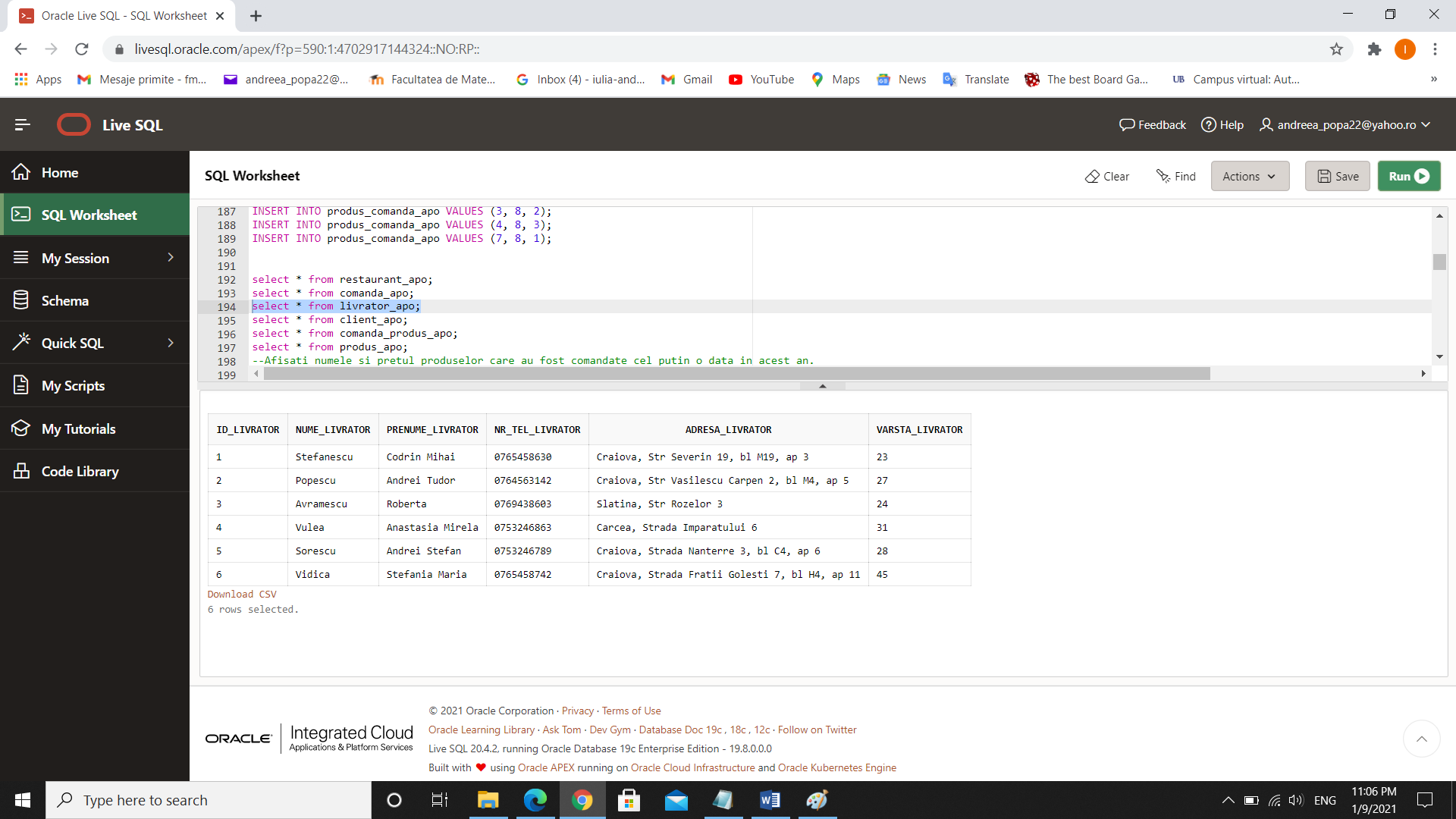
INSERT INTO livrator\_apo VALUES (2, 'Popescu', 'Andrei Tudor', '0764563142', 'Craiova, Str Vasilescu Carpen 2, bl M4, ap 5', '27');

INSERT INTO livrator\_apo VALUES (3, 'Avramescu', 'Roberta', '0769438603', 'Slatina, Str Rozelor 3', '24');

INSERT INTO livrator\_apo VALUES (4, 'Vulea', 'Anastasia Mirela', '0753246863', 'Carcea, Strada Imparatului 6', '31');

INSERT INTO livrator\_apo VALUES (5, 'Sorescu', 'Andrei Stefan', '0753246789', 'Craiova, Strada Nanterre 3, bl C4, ap 6', '28');

INSERT INTO livrator\_apo VALUES (6, 'Vidica', 'Stefania Maria', '0765458742', 'Craiova, Strada Fratii Golesti 7, bl H4, ap 11', '45');



INSERT INTO produs\_apo VALUES (1, 1, 'Snitel de pui', 20, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (2, 1, 'Snitel de porc', 22, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (3, 1, 'Snitel de vita', 25, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (4, 1, 'Cartofi prajiti', 10, 200, 'Garnitura');

INSERT INTO produs\_apo VALUES (5, 1, 'Legume mexicane', 12, 200, 'Garnitura');

INSERT INTO produs\_apo VALUES (6, 1, 'Salata Caesar', 31, 400, 'Salata');

INSERT INTO produs\_apo VALUES (7, 1, 'Coca Cola', 9, 250, 'Bautura racoritoare');

INSERT INTO produs\_apo VALUES (8, 1, 'Sprite', 9, 250, 'Bautura racoritoare');

INSERT INTO produs\_apo VALUES (9, 2, 'Pui la gratar', 23, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (10, 2, 'Cascaval pane', 15, 300, 'Fel principal');

INSERT INTO produs\_apo VALUES (11, 2, 'Ceafa de porc', 23, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (12, 2, 'Piure de mazare', 12, 200, 'Garnitura');

INSERT INTO produs\_apo VALUES (13, 2, 'Cartofi wedges', 12, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (14, 2, 'Salata greceasca', 27, 400, 'Salata');

INSERT INTO produs\_apo VALUES (15, 2, 'Lava cake', 20, 150, 'Desert');

INSERT INTO produs\_apo VALUES (16, 2, 'Creme brulee', 20, 150, 'Desert');

INSERT INTO produs\_apo VALUES (17, 3, 'Crispy de pui', 25, 300, 'Fel principal');

INSERT INTO produs\_apo VALUES (18, 3, 'Snitel de pui', 20, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (19, 3, 'Cotlet de porc', 24, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (20, 3, 'Pastrav la gratar', 28, 300, 'Fel principal');

INSERT INTO produs\_apo VALUES (21, 3, 'Cartofi prajiti', 10, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (22, 3, 'Cartofi la cuptor', 10, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (23, 3, 'Legume mexicane', 10, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (24, 3, 'Vin alb', 12, 150, 'Bautura alcoolica');

INSERT INTO produs\_apo VALUES (25, 3, 'Vin rosu', 12, 150, 'Bautura alcoolica');

INSERT INTO produs\_apo VALUES (26, 3, 'Vin alb', 12, 150, 'Bautura alcoolica');

INSERT INTO produs\_apo VALUES (27, 4, 'Mix bruschete', 18, 200, 'Aperitiv');

INSERT INTO produs\_apo VALUES (28, 4, 'Salata de vinete', 12, 150, 'Aperitiv');

INSERT INTO produs\_apo VALUES (29, 4, 'Bruschete cu somon', 14, 150, 'Aperitiv');

INSERT INTO produs\_apo VALUES (30, 4, 'Parmachef', 30, 350, 'Fel principal');

INSERT INTO produs\_apo VALUES (31, 4, 'Focaccia', 12, 150, 'Extra');

INSERT INTO produs\_apo VALUES (32, 4, 'Sos de usturoi', 4, 100, 'Extra');

INSERT INTO produs\_apo VALUES (33, 4, 'Sos barbeque', 4, 100, 'Extra');

INSERT INTO produs\_apo VALUES (35, 4, 'Tiramisu', 21, 200, 'Desert');

INSERT INTO produs\_apo VALUES (36, 5, 'Pui la gratar', 18, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (37, 5, 'Snitel de pui', 20, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (38, 5, 'Crispy de pui', 22, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (39, 5, 'Parmachef de pui', 28, 300, 'Fel princiapl');

INSERT INTO produs\_apo VALUES (40, 5, 'Cartofi gratinati', 10, 200, 'Garnitura');

INSERT INTO produs\_apo VALUES (41, 5, 'Cartofi prajiti', 10, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (42, 5, 'Legume mexicane', 12, 250, 'Garnitura');

INSERT INTO produs\_apo VALUES (43, 5, 'Cidru de mere', 9, 250, 'Bautura alcoolica');

INSERT INTO produs\_apo VALUES (44, 5, 'Sprite', 8, 250, 'Bautura racoritoare');

INSERT INTO produs\_apo VALUES (45, 5, 'Sos sour', 4, 100, 'Extra');

INSERT INTO produs\_apo VALUES (46, 5, 'Tiramisu', 15, 200, 'Desert');

INSERT INTO produs\_apo VALUES (47, 6, 'Pui la gratar', 16, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (48, 6, 'Salata Caesar', 24, 300, 'Salata');

INSERT INTO produs\_apo VALUES (49, 6, 'Paste Carbonara', 28, 300, 'Paste');

INSERT INTO produs\_apo VALUES (50, 6, 'Paste bollognese', 26, 300, 'Paste');

INSERT INTO produs\_apo VALUES (51, 6, 'Tiramisu', 20, 200, 'Desert');

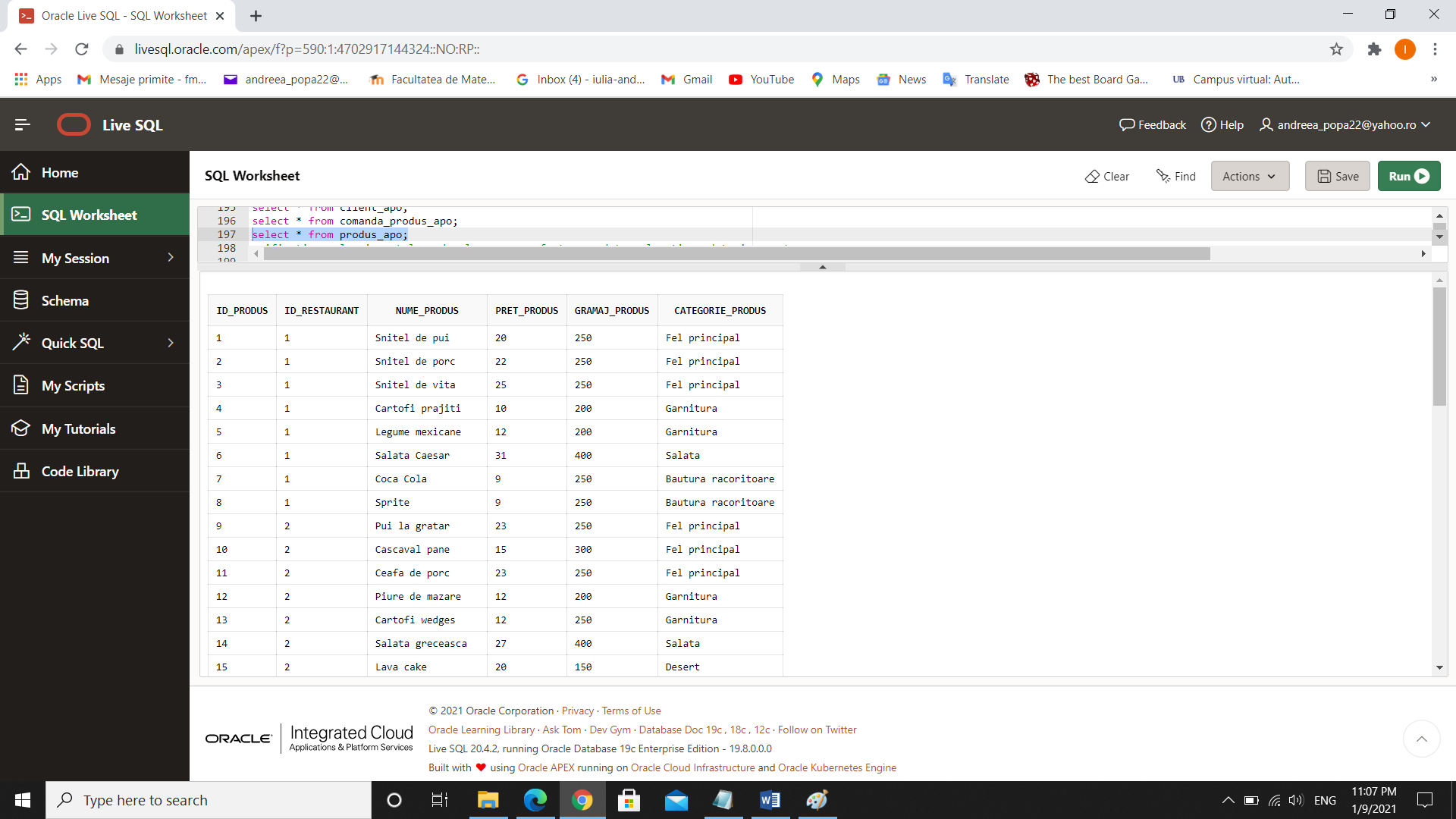
INSERT INTO produs\_apo VALUES (52, 6, 'Sos barbeque', 5, 100, 'Extra');

INSERT INTO produs\_apo VALUES (53, 6, 'Crispy de pui', 23, 250, 'Fel principal');

INSERT INTO produs\_apo VALUES (54, 6, 'Cartofi prajiti', 12, 200, 'Garnitura');

INSERT INTO produs\_apo VALUES (55, 6, 'Sprite', 10, 250, 'Bautura racoritoare');

INSERT INTO produs\_apo VALUES (56, 6, 'Cidru de mere', 12, 250, 'Bautura alcoolica');



INSERT INTO client\_apo VALUES (1, 'Popa', 'Iulia Andreea', 'Craiova, Str Vasilescu Carpen 2, bl M15, ap 6', '0756548535', 'andreea\_popa22@yahoo.ro', 20);

INSERT INTO client\_apo VALUES (2, 'Badescu', 'Gabriel', 'Pitesti, Str Eroilor 5, bl C4, ap 6', '0768546830', 'gabri16@gmail.com', 20);

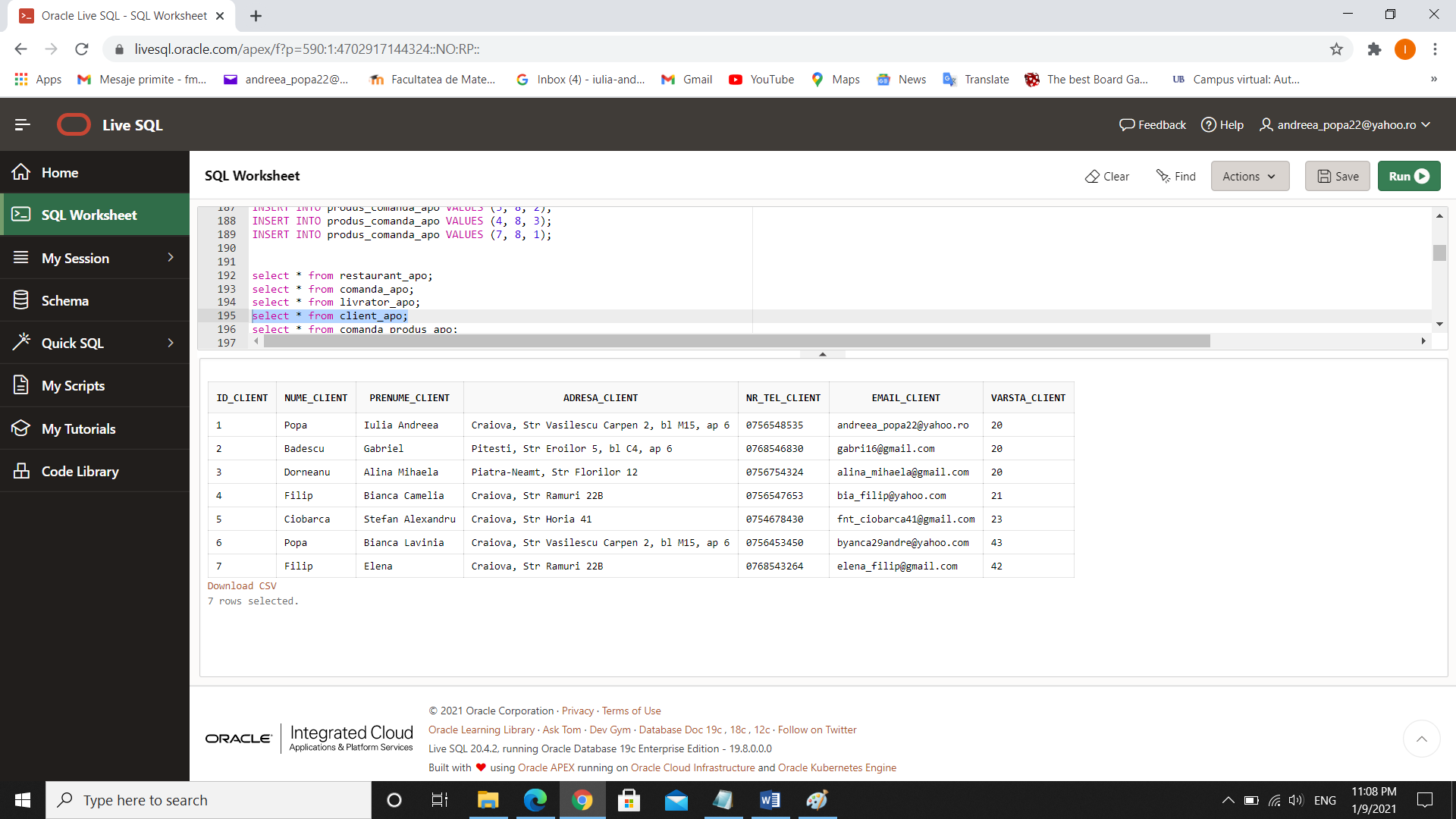
INSERT INTO client\_apo VALUES (3, 'Dorneanu', 'Alina Mihaela', 'Piatra-Neamt, Str Florilor 12', '0756754324', 'alina\_mihaela@gmail.com', 20);

INSERT INTO client\_apo VALUES (4, 'Filip', 'Bianca Camelia', 'Craiova, Str Ramuri 22B', '0756547653', 'bia\_filip@yahoo.com', 21);

INSERT INTO client\_apo VALUES (5, 'Ciobarca', 'Stefan Alexandru', 'Craiova, Str Horia 41', '0754678430', 'fnt\_ciobarca41@gmail.com', 23);

INSERT INTO client\_apo VALUES (6, 'Popa', 'Bianca Lavinia', 'Craiova, Str Vasilescu Carpen 2, bl M15, ap 6', '0756453450', 'byanca29andre@yahoo.com', 43);

INSERT INTO client\_apo VALUES (7, 'Filip', 'Elena', 'Craiova, Str Ramuri 22B', '0768543264', 'elena\_filip@gmail.com', 42);



INSERT INTO comanda\_apo VALUES (1, 1, 2, 3, 129, '21-JAN-06');

INSERT INTO produs\_comanda\_apo VALUES (17, 1, 1);

INSERT INTO produs\_comanda\_apo VALUES (19, 1, 1);

INSERT INTO produs\_comanda\_apo VALUES (18, 1, 1);

INSERT INTO produs\_comanda\_apo VALUES (21, 1, 3);

INSERT INTO produs\_comanda\_apo VALUES (23, 1, 3);

INSERT INTO comanda\_apo VALUES (2, 6, 2, 3, 90, '21-JAN-05');

INSERT INTO produs\_comanda\_apo VALUES (18, 2, 1);

INSERT INTO produs\_comanda\_apo VALUES (20, 2, 1);

INSERT INTO produs\_comanda\_apo VALUES (24, 2, 1);

INSERT INTO produs\_comanda\_apo VALUES (22, 2, 2);

INSERT INTO comanda\_apo VALUES (3, 2, 3, 4, 90, '21-JAN-04');

INSERT INTO produs\_comanda\_apo VALUES (30, 3, 2);

INSERT INTO produs\_comanda\_apo VALUES (31, 3, 1);

INSERT INTO produs\_comanda\_apo VALUES (32, 3, 1);

INSERT INTO produs\_comanda\_apo VALUES (33, 3, 2);

INSERT INTO comanda\_apo VALUES (4, 1, 3, 5, 125, '20-DEC-23');

INSERT INTO produs\_comanda\_apo VALUES (38, 4, 1);

INSERT INTO produs\_comanda\_apo VALUES (39, 4, 1);

INSERT INTO produs\_comanda\_apo VALUES (40, 4, 2);

INSERT INTO produs\_comanda\_apo VALUES (43, 4, 2);

INSERT INTO produs\_comanda\_apo VALUES (46, 4, 2);

INSERT INTO comanda\_apo VALUES (5, 3, 2, 1, 115, '20-DEC-22');

INSERT INTO produs\_comanda\_apo VALUES (1, 5, 1);

INSERT INTO produs\_comanda\_apo VALUES (3, 5, 2);

INSERT INTO produs\_comanda\_apo VALUES (4, 5, 3);

INSERT INTO produs\_comanda\_apo VALUES (7, 5, 1);

INSERT INTO comanda\_apo VALUES (6, 1, 4, 3, 140, '21-JAN-07');

INSERT INTO produs\_comanda\_apo VALUES (17, 6, 4);

INSERT INTO produs\_comanda\_apo VALUES (21, 6, 4);

INSERT INTO comanda\_apo VALUES (7, 3, 2, 1, 115, '20-DEC-22');

INSERT INTO produs\_comanda\_apo VALUES (1, 7, 1);

INSERT INTO produs\_comanda\_apo VALUES (3, 7, 2);

INSERT INTO produs\_comanda\_apo VALUES (4, 7, 3);

INSERT INTO produs\_comanda\_apo VALUES (7, 7, 1);

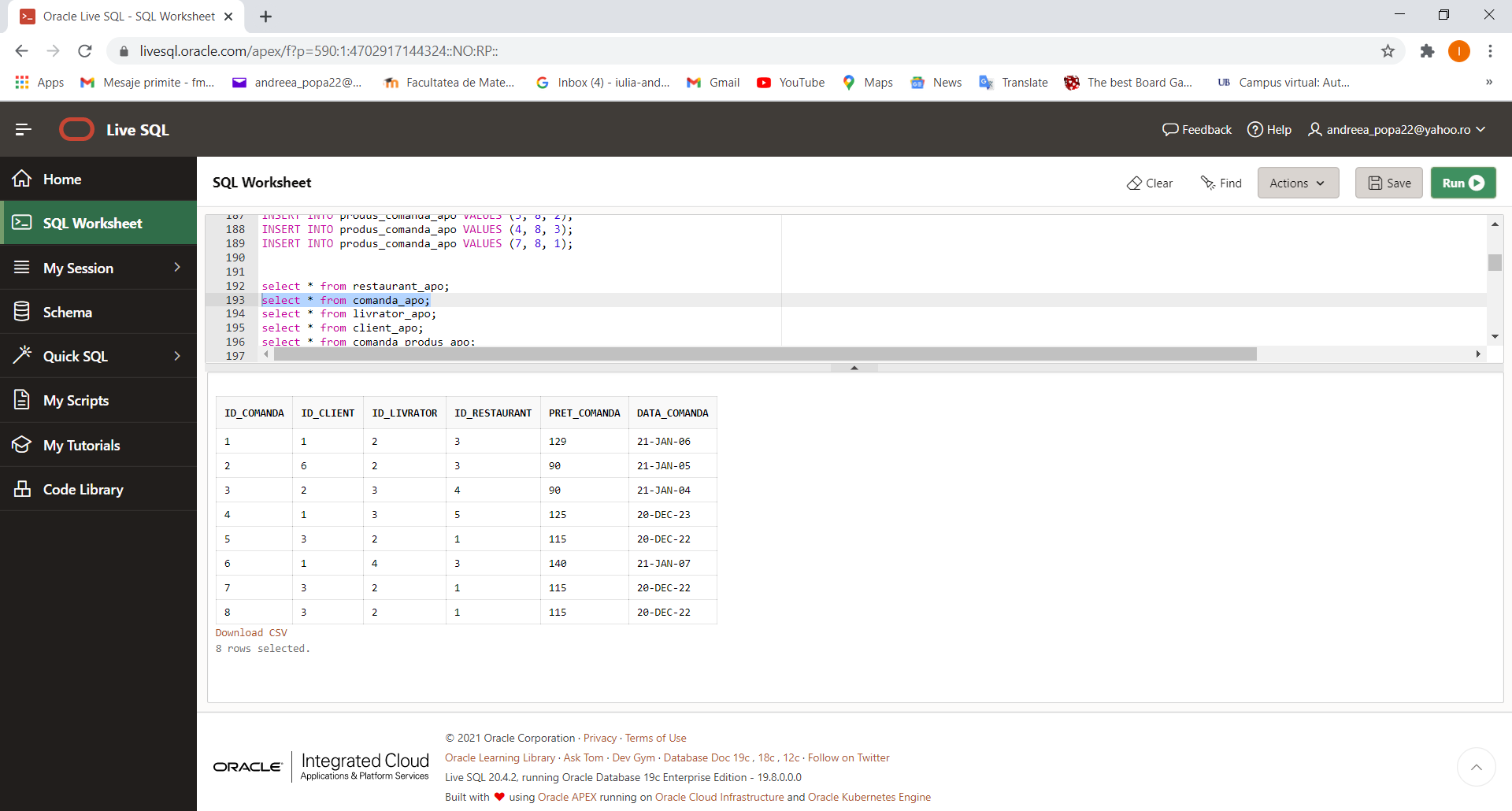
INSERT INTO comanda\_apo VALUES (8, 3, 2, 1, 115, '20-DEC-22');

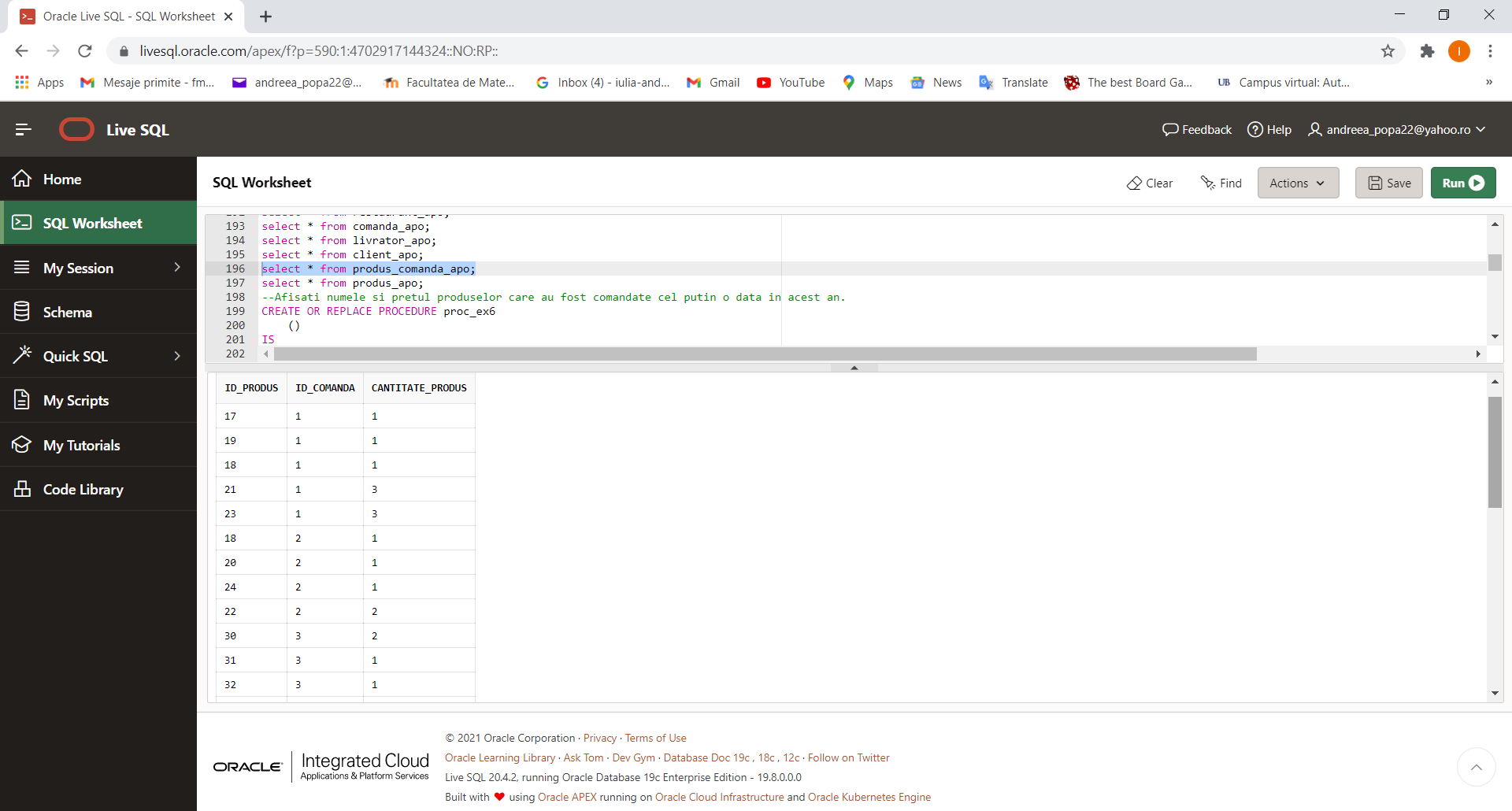
INSERT INTO produs\_comanda\_apo VALUES (1, 8, 1);

INSERT INTO produs\_comanda\_apo VALUES (3, 8, 2);

INSERT INTO produs\_comanda\_apo VALUES (4, 8, 3);

INSERT INTO produs\_comanda\_apo VALUES (7, 8, 1);





1. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

--6. Pentru fiecare client (nume, prenume) cu varsta mai mare decat un numar dat de la tastatura, afisati numele

--restaurantului la care a plasat cele mai multe comenzi.

CREATE OR REPLACE PROCEDURE proc\_ex6

(idd1 IN OUT NUMBER)

IS

TYPE tab\_ind\_comenzi IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;

t1 tab\_ind\_comenzi;

idr comanda\_apo.id\_restaurant%TYPE := NULL;

BEGIN

SELECT id\_restaurant

BULK COLLECT INTO t1

FROM (SELECT COUNT(\*) AS num, id\_restaurant

FROM comanda\_apo c, client\_apo cl

WHERE c.id\_client = cl.id\_client

AND c.id\_client = idd1

GROUP BY id\_restaurant) y

order by y.num desc;

idd1 := t1(1);

END;

/

DECLARE

TYPE rec IS RECORD (id1 client\_apo.id\_client%TYPE,

nume client\_apo.nume\_client%TYPE,

prenume client\_apo.prenume\_client%TYPE);

TYPE tab\_ind\_clienti IS TABLE OF rec INDEX BY PLS\_INTEGER;

t tab\_ind\_clienti;

age client\_apo.varsta\_client%TYPE := '19';

idd client\_apo.id\_client%TYPE := NULL;

rest restaurant\_apo.nume\_rest%TYPE;

BEGIN

SELECT distinct c.id\_client, c.nume\_client, c.prenume\_client

BULK COLLECT INTO t

FROM client\_apo c RIGHT JOIN comanda\_apo cm

ON c.id\_client = cm.id\_client

WHERE varsta\_client > age

order by c.id\_client; --punem in tabelul indexat clientii care au plasat cel putin o comanda si care au varsta mai mare decat numarul dat de la tastatura

FOR i IN t.FIRST..t.LAST LOOP

idd := t(i).id1;

proc\_ex6(idd);

SELECT nume\_rest

INTO rest

FROM restaurant\_apo

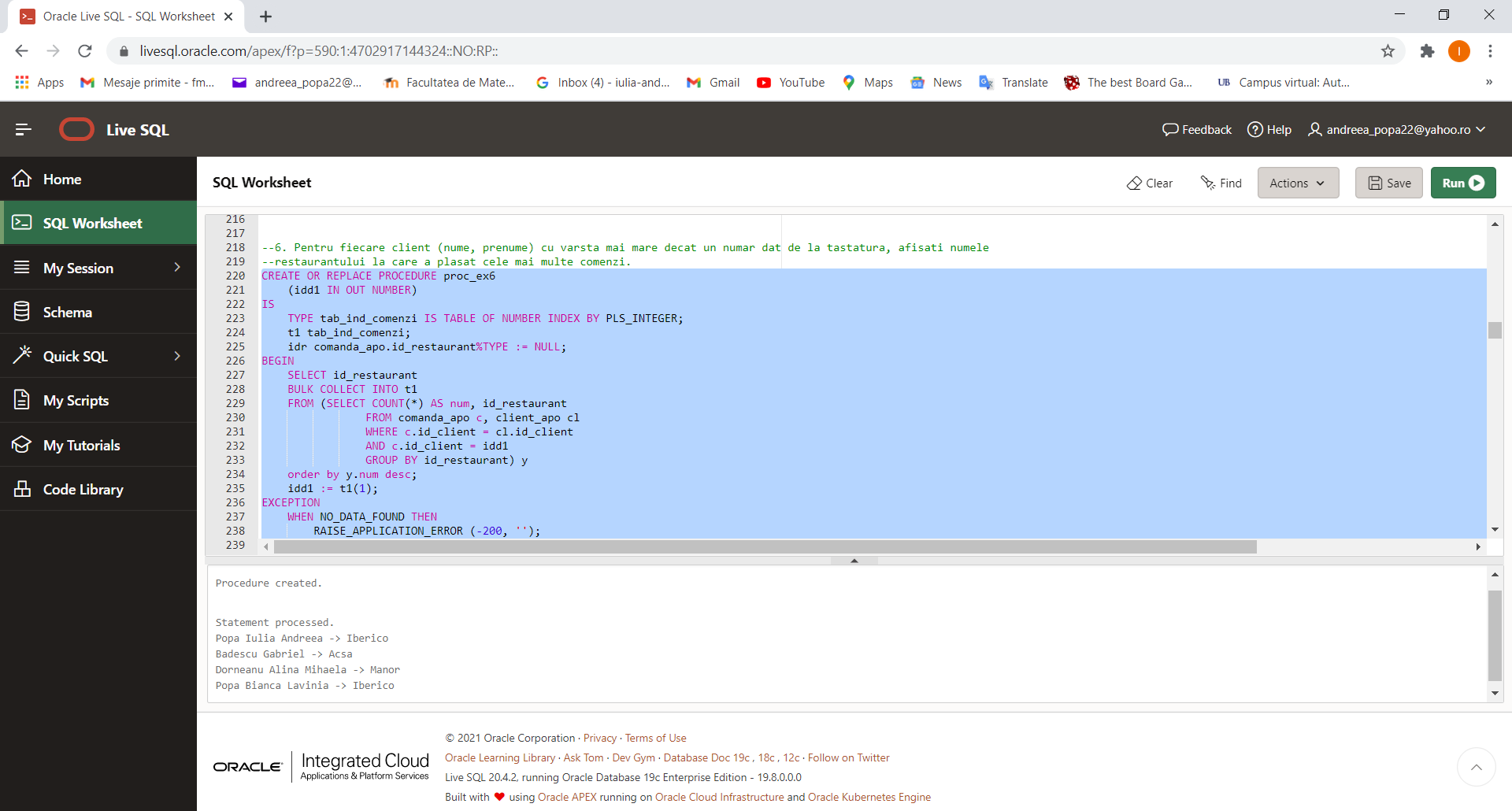
WHERE id\_restaurant = idd;

DBMS\_OUTPUT.PUT\_LINE(t(i).nume || ' ' || t(i).prenume || ' -> ' || rest);

END LOOP;

END;

/



1. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

--7. Pentru restaurant cu un id dat, afisati numele produselor care au fost comandate cel putin o data si de cate ori. (ciclu cursor cu subcereri)

CREATE OR REPLACE PROCEDURE proc\_ex7

(idd IN NUMBER)

IS

nume produs\_apo.nume\_produs%TYPE;

BEGIN

FOR i IN (SELECT p.id\_produs idp, SUM(pc.cantitate\_produs) cantitate

from produs\_apo p, produs\_comanda\_apo pc

WHERE p.id\_produs = pc.id\_produs

AND p.id\_restaurant = idd

group by p.id\_produs) LOOP

SELECT nume\_produs

into nume

from produs\_apo

where id\_produs = i.idp;

DBMS\_OUTPUT.PUT\_LINE(nume || ' ' || i.cantitate);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR (-200, '');

END;

/

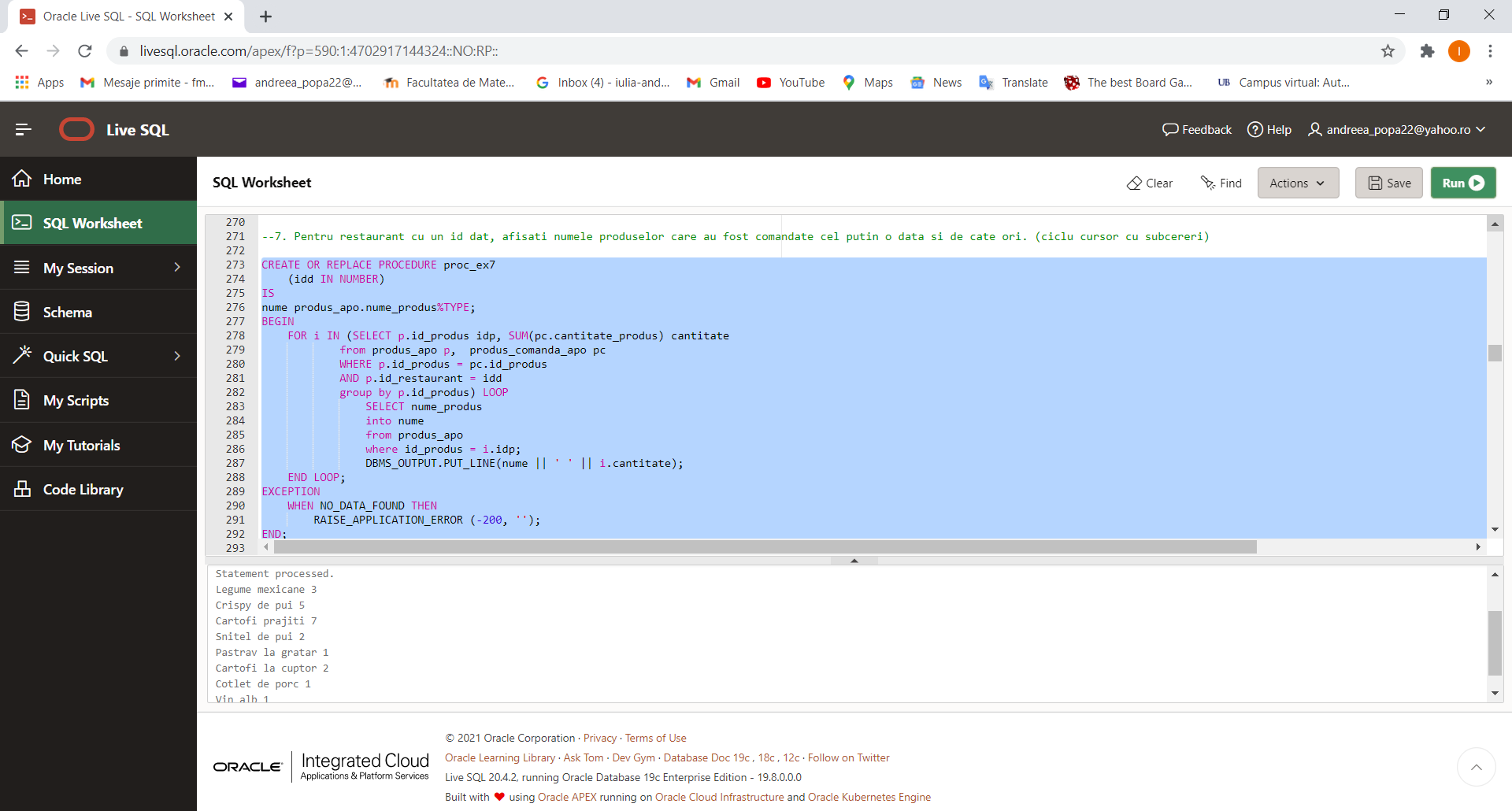
DECLARE

idr restaurant\_apo.id\_restaurant%TYPE := 3;

BEGIN

proc\_ex7(idr);

END;



1. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--8. Pentru fiecare livrator(nume, prenume), afisati numele si prenumele clientului la care a mers cel mai des si cate comenzi i-a livrat.

CREATE OR REPLACE FUNCTION fct\_ex8

(idd NUMBER)

RETURN number

IS

exceptie EXCEPTION;

rezultat number := 0;

TYPE rec IS RECORD (idd livrator\_apo.id\_livrator%TYPE,

nume client\_apo.nume\_client%TYPE,

prenume client\_apo.prenume\_client%TYPE,

nr number(4));

TYPE tab\_ind\_clienti IS TABLE OF rec INDEX BY PLS\_INTEGER;

t tab\_ind\_clienti;

BEGIN

SELECT l.id\_livrator, cl.nume\_client, cl.prenume\_client, COUNT(cl.nume\_client) NR

BULK COLLECT INTO t

from livrator\_apo l inner join comanda\_apo c on l.id\_livrator = c.id\_livrator inner join client\_apo cl on c.id\_client = cl.id\_client

where l.id\_livrator = idd

group by l.id\_livrator, cl.nume\_client, cl.prenume\_client

order by NR desc;

IF t.COUNT() = 0 THEN RAISE exceptie;

ELSE

DBMS\_OUTPUT.PUT\_LINE(' a livrat la ' || t(1).nume || ' ' || t(1).prenume);

rezultat := t(1).nr;

return rezultat;

end if;

EXCEPTION

when exceptie then

DBMS\_OUTPUT.PUT('nu a livrat nicio comanda.');

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

return -1;

END;

/

DECLARE

v\_id livrator\_apo.id\_livrator%TYPE;

v\_nume livrator\_apo.nume\_livrator%TYPE;

v\_prenume livrator\_apo.prenume\_livrator%TYPE;

CURSOR c IS

SELECT id\_livrator, nume\_livrator, prenume\_livrator

FROM livrator\_apo;

rez VARCHAR2(150);

BEGIN

OPEN c;

LOOP

FETCH c INTO v\_id, v\_nume, v\_prenume;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT(v\_nume || ' ' || v\_prenume || ' ');

rez := fct\_ex8(v\_id);

IF rez <> -1 THEN

DBMS\_OUTPUT.PUT\_LINE( rez || ' comenzi');

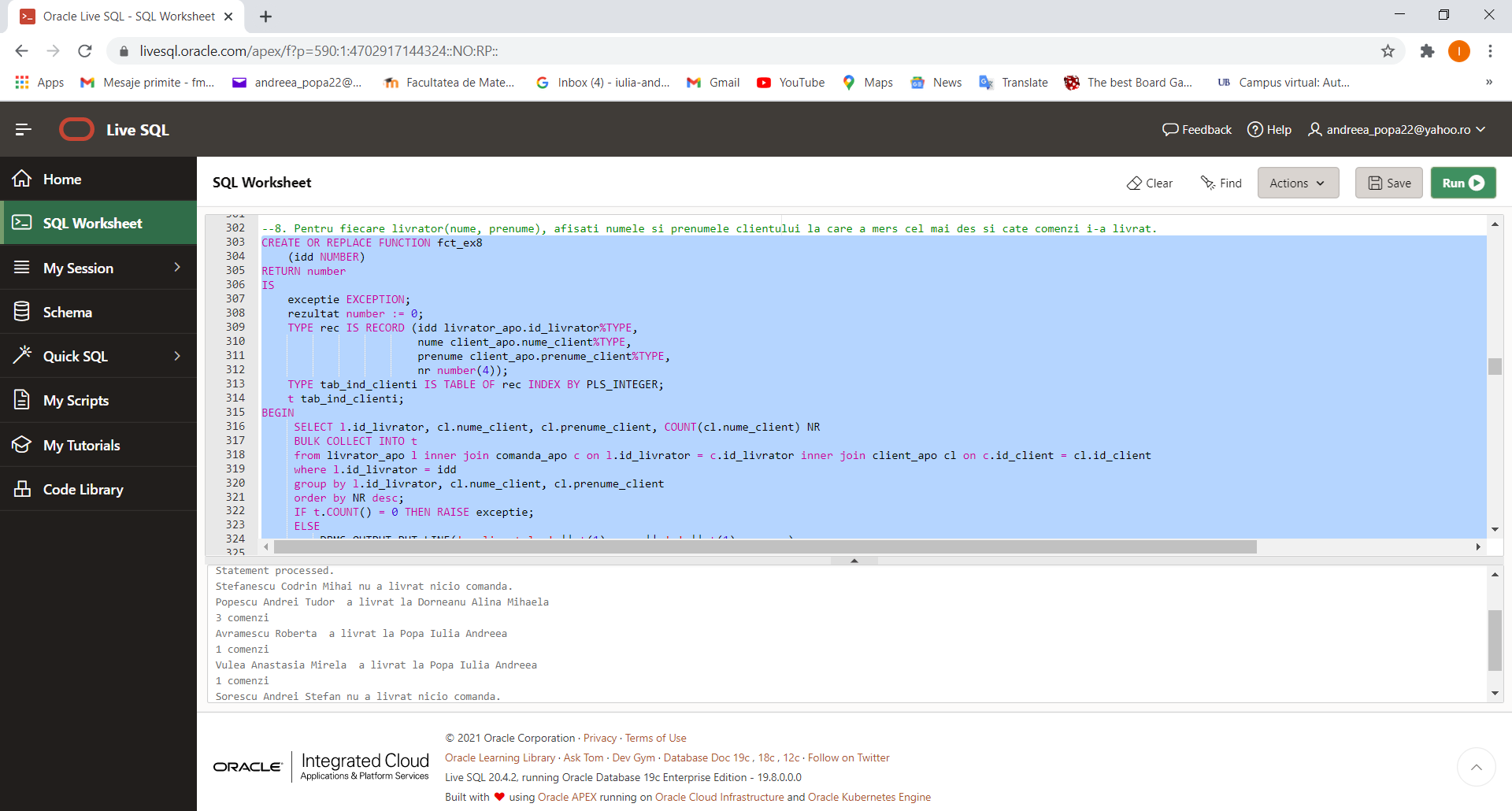
END IF;

END LOOP;

CLOSE c;

END;

/



1. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--9. Pentru toate comenzile plasate pe o data anume, afisati toate detaliile acesteia( nume,

--prenume livrator; produse comandate; nume restaurant; nume, prenume client)

CREATE OR REPLACE PROCEDURE proc\_ex9

(idd IN number)

IS

TYPE rec IS RECORD (id\_com comanda\_apo.id\_comanda%TYPE,

nume\_rest restaurant\_apo.nume\_rest%TYPE,

nume\_cl client\_apo.nume\_client%TYPE,

prenume\_cl client\_apo.prenume\_client%TYPE,

nume\_liv livrator\_apo.nume\_livrator%TYPE,

prenume\_liv livrator\_apo.prenume\_livrator%TYPE);

r rec;

TYPE tab\_ind IS TABLE OF produs\_apo.nume\_produs%TYPE INDEX BY PLS\_INTEGER;

t tab\_ind;

BEGIN

select c.id\_comanda, r.nume\_rest, cl.nume\_client, cl.prenume\_client, l.nume\_livrator, l.prenume\_livrator

into r

from restaurant\_apo r join comanda\_apo c on r.id\_restaurant = c.id\_restaurant join livrator\_apo l on c.id\_livrator = l.id\_livrator join client\_apo cl on c.id\_client = cl.id\_client

where id\_comanda = idd;

DBMS\_OUTPUT.PUT\_LINE('Comanda cu id-ul ' || r.id\_com || ' a fost plasata de ' || r.nume\_cl ||' '||r.prenume\_cl||' la '||r.nume\_rest||' si livrata de '||r.nume\_liv||' '||r.prenume\_liv||' si contine urmatoarele produse: ');

select p.nume\_produs

bulk collect into t

from produs\_apo p join produs\_comanda\_apo pc on p.id\_produs = pc.id\_produs join comanda\_apo c on c.id\_comanda = pc.id\_comanda

where c.id\_comanda = idd;

for i in t.first()..t.last() loop

DBMS\_OUTPUT.PUT(t(i) || ' ');

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

end loop;

END;

/

DECLARE

v\_data DATE := '20-DEC-02';

i number := -1;

exceptie1 exception;

BEGIN

FOR i IN (SELECT id\_comanda

from comanda\_apo

where data\_comanda = v\_data) LOOP

proc\_ex9(i.id\_comanda);

END LOOP;

if i = -1 then raise exceptie1;

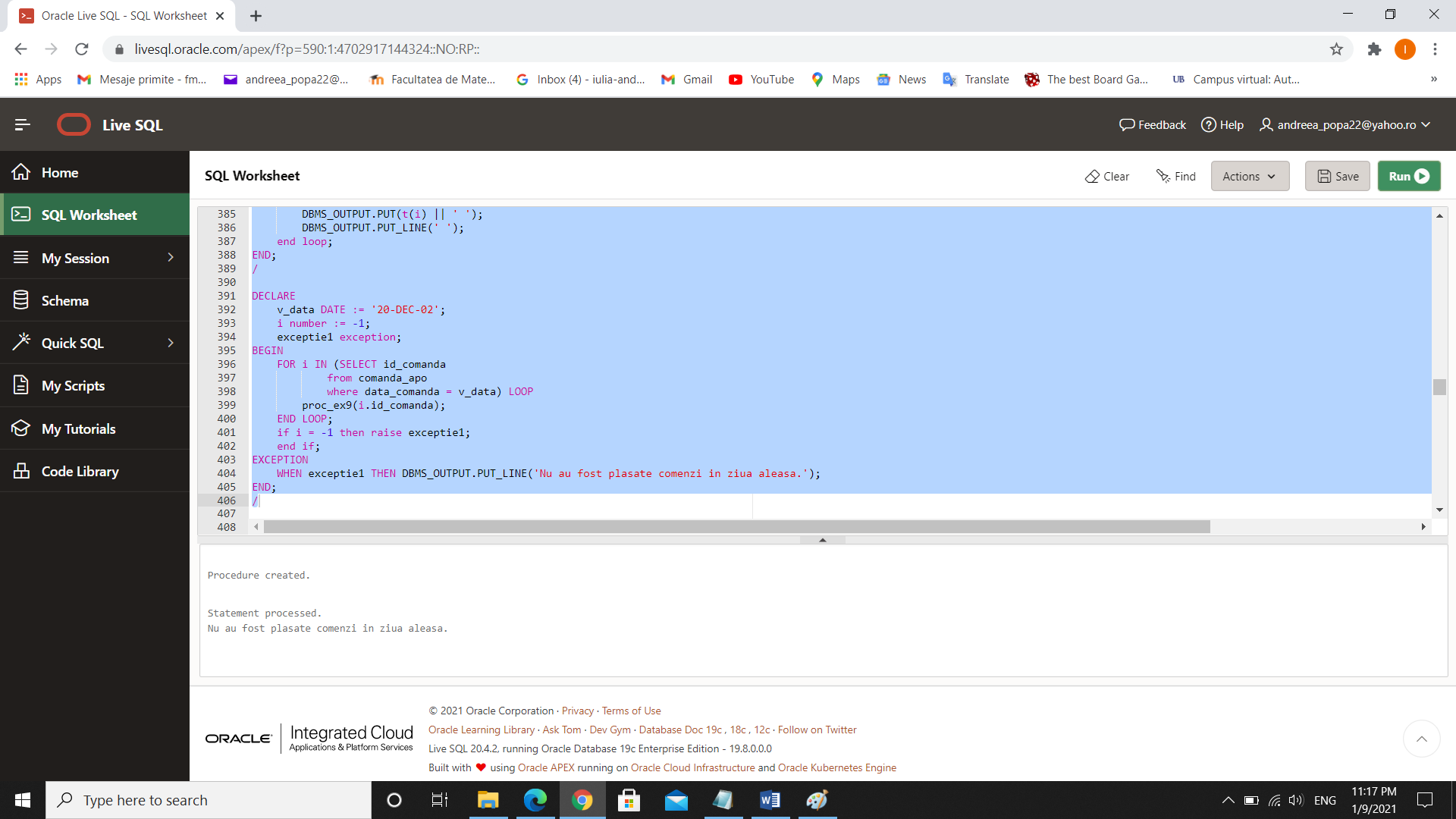
end if;

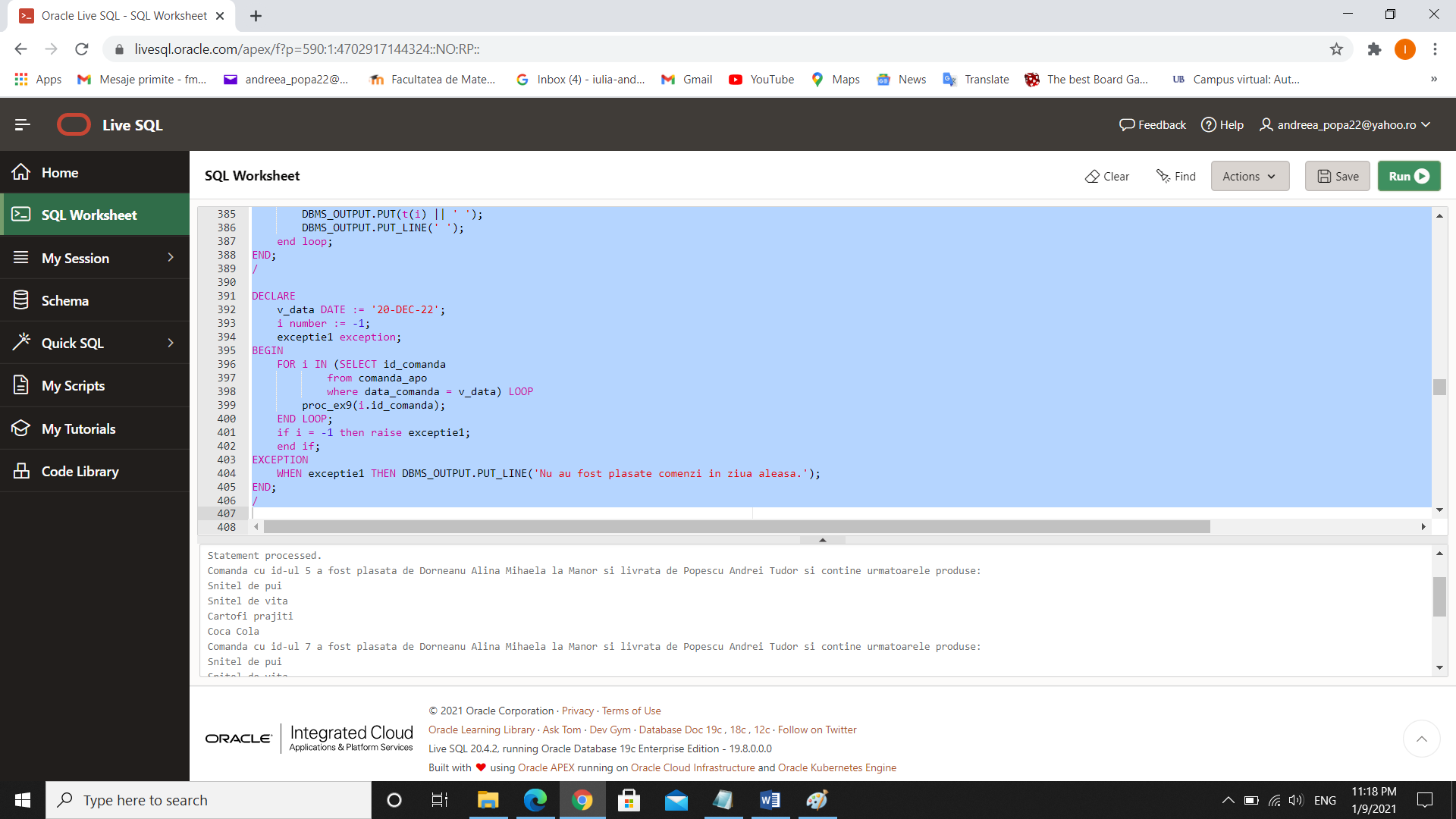
EXCEPTION

WHEN exceptie1 THEN DBMS\_OUTPUT.PUT\_LINE('Nu au fost plasate comenzi in ziua aleasa.');

END;

/





1. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

--10. Definiti un trigger care sa nu permita plasarea/editarea/stergerea comenzilor miercurea de la 17 la 19 (interval de mentenanta).

CREATE OR REPLACE TRIGGER trig\_ex10

BEFORE INSERT OR DELETE OR UPDATE on comanda\_apo

BEGIN

IF(TO\_CHAR(SYSDATE,'D') = 4) AND (TO\_CHAR(SYSDATE,'HH24') BETWEEN 17 AND 19) THEN

IF INSERTING THEN

RAISE\_APPLICATION\_ERROR(-20001,'Plasarea comenzilor nu este posibila in intervalul destinat mentenantei bazei de date.');

ELSIF DELETING THEN

RAISE\_APPLICATION\_ERROR(-20002,'Stergea comenzilor nu este posibila in intervalul destinat mentenantei bazei de date.');

ELSE

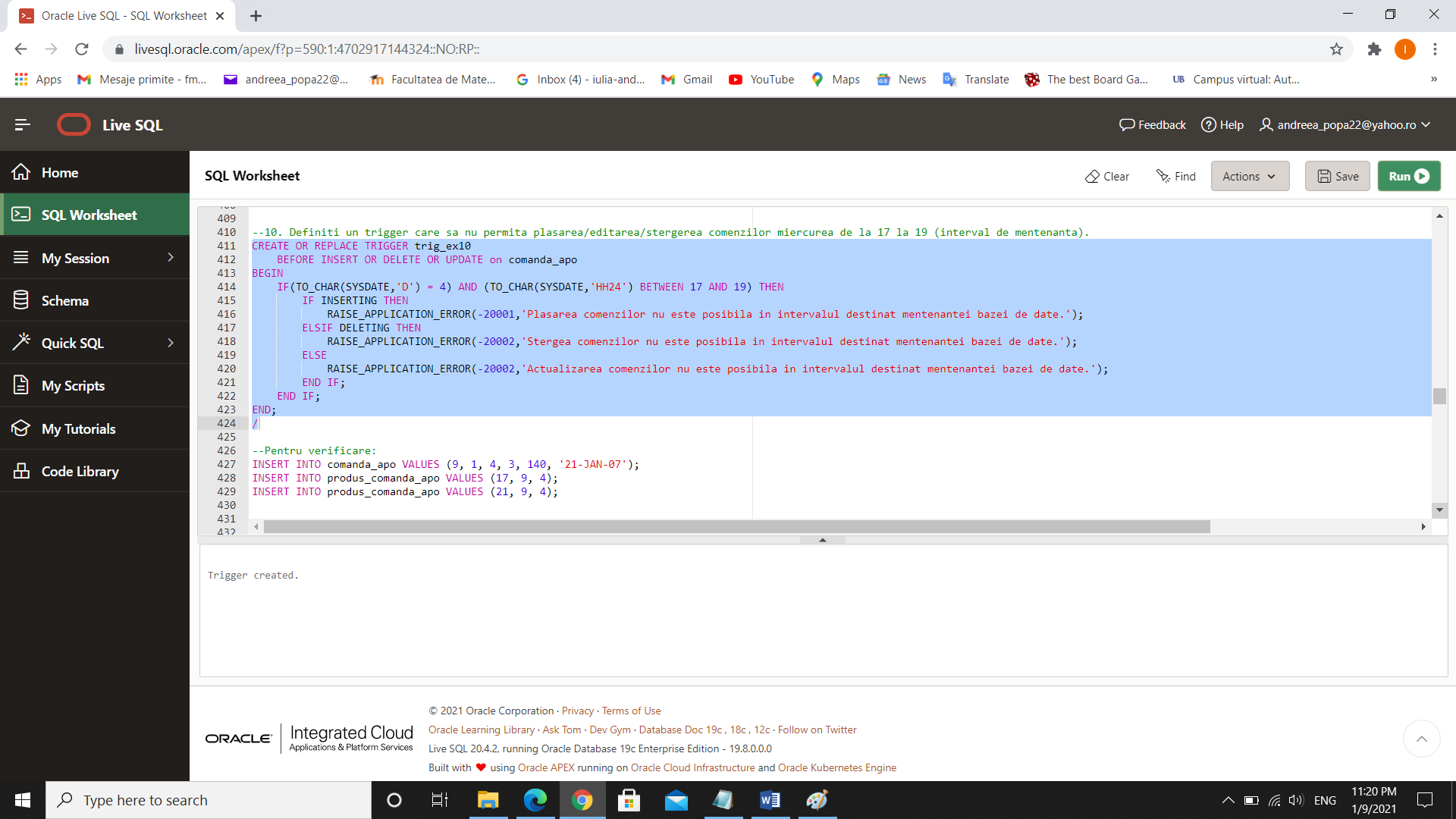
RAISE\_APPLICATION\_ERROR(-20002,'Actualizarea comenzilor nu este posibila in intervalul destinat mentenantei bazei de date.');

END IF;

END IF;

END;

/

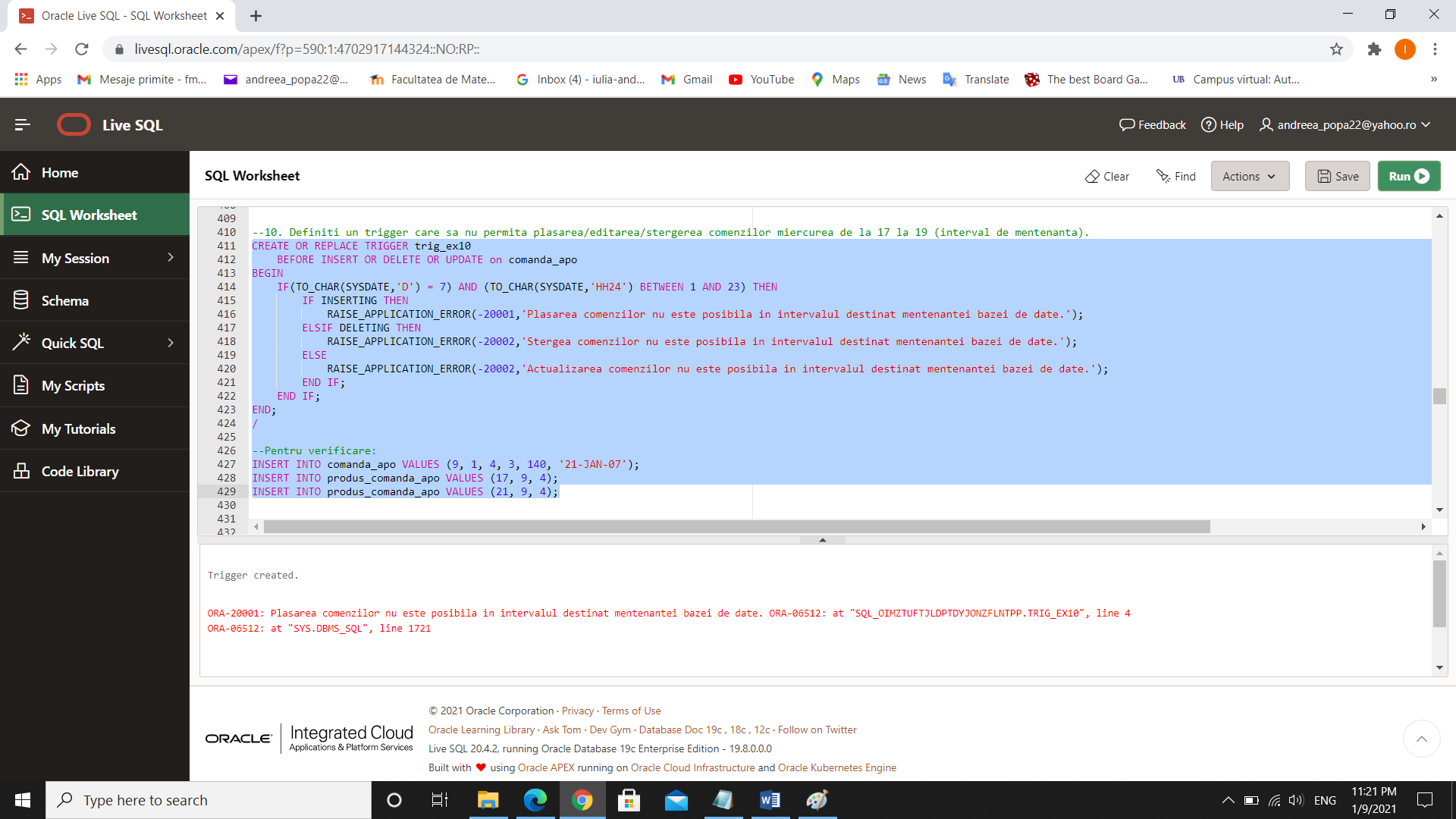


--Pentru verificare:

INSERT INTO comanda\_apo VALUES (9, 1, 4, 3, 140, '21-JAN-07');

INSERT INTO produs\_comanda\_apo VALUES (17, 9, 4);

INSERT INTO produs\_comanda\_apo VALUES (21, 9, 4);



1. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

--11. Definiti un trigger care sa nu permita micsorarea pretului unei comenzi cu valoare mai mica de 100 de lei.

--(pentru acestea nu sunt disponibile reduceri)

CREATE OR REPLACE PROCEDURE f\_reducere

(v\_id number)

IS

BEGIN

UPDATE comanda\_apo

SET pret\_comanda = pret\_comanda - 10

WHERE id\_comanda = v\_id;

END;

/

CREATE OR REPLACE TRIGGER trig\_ex11

BEFORE UPDATE of pret\_comanda on comanda\_apo

FOR EACH ROW

BEGIN

IF :OLD.pret\_comanda < 100 AND :NEW.pret\_comanda < :OLD.pret\_comanda THEN

RAISE\_APPLICATION\_ERROR(-20004,'Comenzile cu valoare mai mica de 100 de lei nu pot beneficia de reduceri.');

END IF;

END;

/

DECLARE

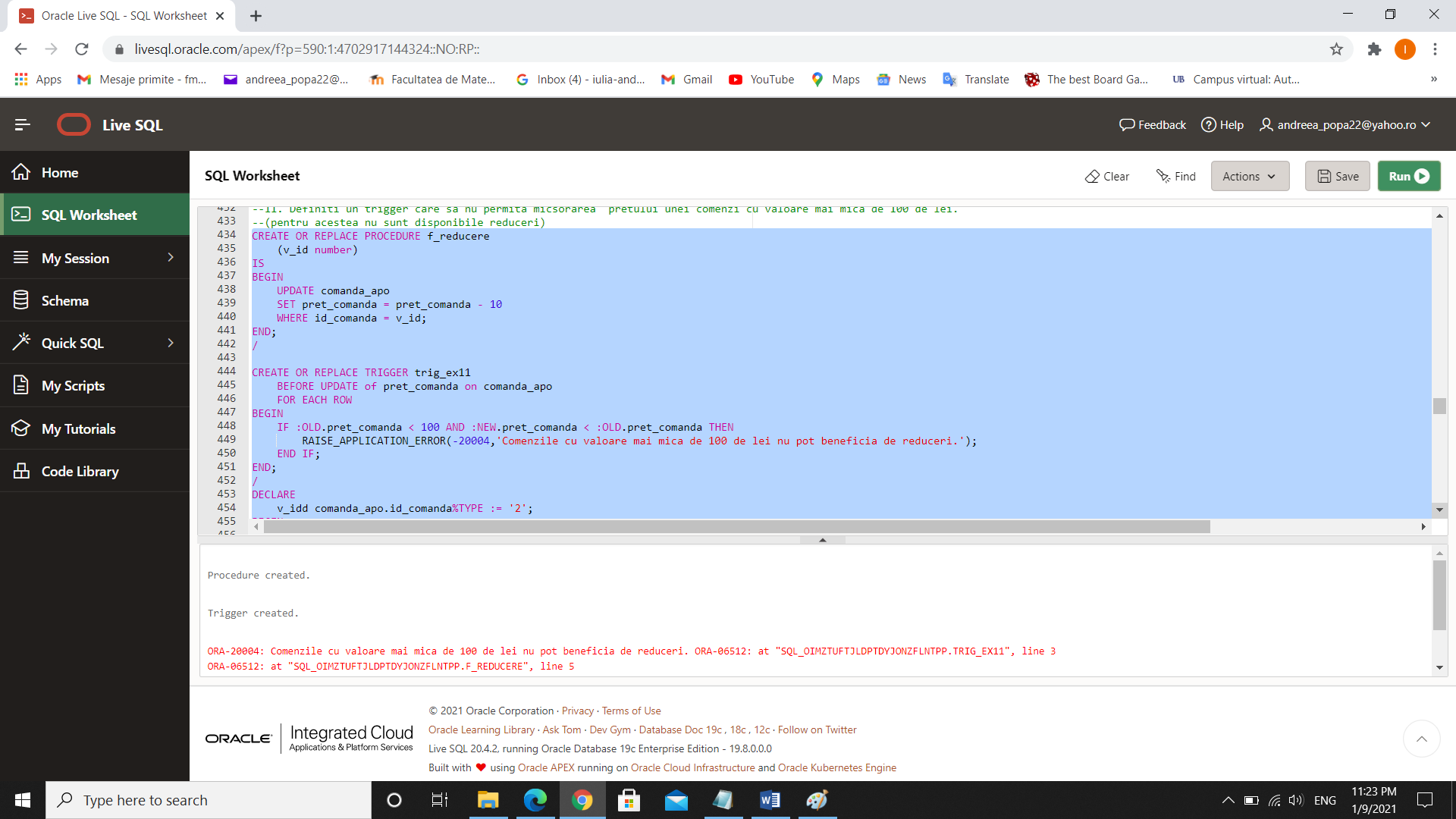
v\_idd comanda\_apo.id\_comanda%TYPE := '2';

BEGIN

f\_reducere(v\_idd);

END;

/



1. Definiți un trigger de tip LDD. Declanșați trigger-ul.

--12. Definiți un declanșator care să introducă date în acest tabel după ce utilizatorul a folosit o comandă LDD

CREATE TABLE audit\_user

(eveniment VARCHAR2(20),

tip\_obiect\_referit VARCHAR2(30),

nume\_obiect\_referit VARCHAR2(30),

data TIMESTAMP(3));

CREATE OR REPLACE TRIGGER audit\_schema

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO audit\_user

VALUES (SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_TYPE,

SYS.DICTIONARY\_OBJ\_NAME, SYSTIMESTAMP(3));

END;

/

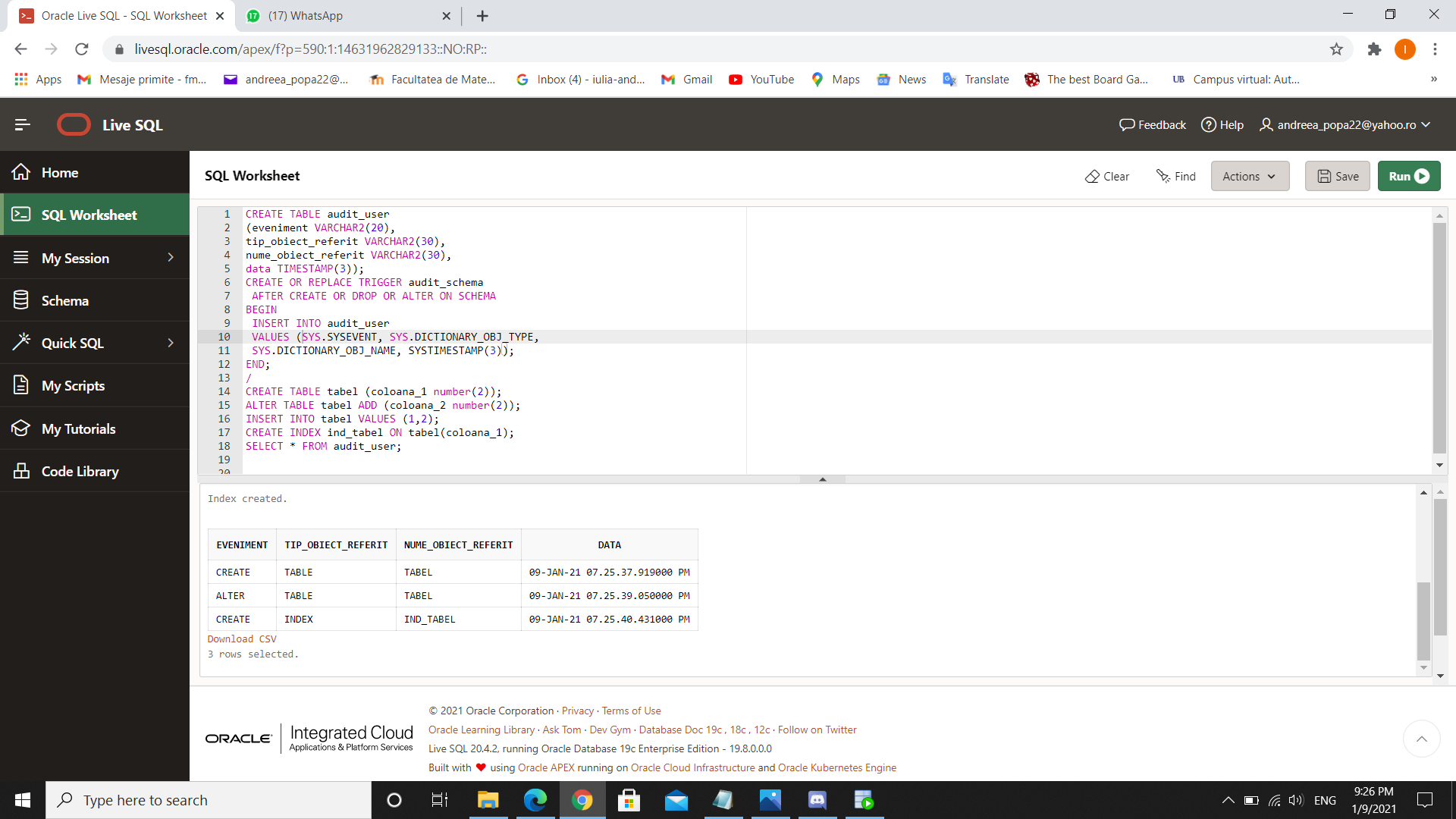
CREATE TABLE tabel (coloana\_1 number(2));

ALTER TABLE tabel ADD (coloana\_2 number(2));

INSERT INTO tabel VALUES (1,2);

CREATE INDEX ind\_tabel ON tabel(coloana\_1);

SELECT \* FROM audit\_user;



1. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

CREATE OR REPLACE PACKAGE pachet1\_apo AS

PROCEDURE proc\_ex6 (idd1 IN OUT number);

PROCEDURE proc\_ex7 (idd number);

FUNCTION fct\_ex8 (idd number) return number;

PROCEDURE proc\_ex9 (idd number);

END pachet1\_apo;

/

CREATE OR REPLACE PACKAGE BODY pachet1\_apo AS

PROCEDURE proc\_ex6

(idd1 IN OUT NUMBER)

IS

TYPE tab\_ind\_comenzi IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;

t1 tab\_ind\_comenzi;

idr comanda\_apo.id\_restaurant%TYPE := NULL;

BEGIN

SELECT id\_restaurant

BULK COLLECT INTO t1

FROM (SELECT COUNT(\*) AS num, id\_restaurant

FROM comanda\_apo c, client\_apo cl

WHERE c.id\_client = cl.id\_client

AND c.id\_client = idd1

GROUP BY id\_restaurant) y

order by y.num desc;

idd1 := t1(1);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR (-200, '');

END proc\_ex6;

PROCEDURE proc\_ex7

(idd IN NUMBER)

IS

nume produs\_apo.nume\_produs%TYPE;

BEGIN

FOR i IN (SELECT p.id\_produs idp, SUM(pc.cantitate\_produs) cantitate

from produs\_apo p, produs\_comanda\_apo pc

WHERE p.id\_produs = pc.id\_produs

AND p.id\_restaurant = idd

group by p.id\_produs) LOOP

SELECT nume\_produs

into nume

from produs\_apo

where id\_produs = i.idp;

DBMS\_OUTPUT.PUT\_LINE(nume || ' ' || i.cantitate);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR (-200, '');

END proc\_ex7;

FUNCTION fct\_ex8

(idd NUMBER)

RETURN number

IS

exceptie EXCEPTION;

rezultat number := 0;

TYPE rec IS RECORD (idd livrator\_apo.id\_livrator%TYPE,

nume client\_apo.nume\_client%TYPE,

prenume client\_apo.prenume\_client%TYPE,

nr number(4));

TYPE tab\_ind\_clienti IS TABLE OF rec INDEX BY PLS\_INTEGER;

t tab\_ind\_clienti;

BEGIN

SELECT l.id\_livrator, cl.nume\_client, cl.prenume\_client, COUNT(cl.nume\_client) NR

BULK COLLECT INTO t

from livrator\_apo l inner join comanda\_apo c on l.id\_livrator = c.id\_livrator inner join client\_apo cl on c.id\_client = cl.id\_client

where l.id\_livrator = idd

group by l.id\_livrator, cl.nume\_client, cl.prenume\_client

order by NR desc;

IF t.COUNT() = 0 THEN RAISE exceptie;

ELSE

DBMS\_OUTPUT.PUT\_LINE(' a livrat la ' || t(1).nume || ' ' || t(1).prenume);

rezultat := t(1).nr;

return rezultat;

end if;

EXCEPTION

when exceptie then

DBMS\_OUTPUT.PUT('nu a livrat nicio comanda.');

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

return -1;

END fct\_ex8;

PROCEDURE proc\_ex9

(idd IN number)

IS

TYPE rec IS RECORD (id\_com comanda\_apo.id\_comanda%TYPE,

nume\_rest restaurant\_apo.nume\_rest%TYPE,

nume\_cl client\_apo.nume\_client%TYPE,

prenume\_cl client\_apo.prenume\_client%TYPE,

nume\_liv livrator\_apo.nume\_livrator%TYPE,

prenume\_liv livrator\_apo.prenume\_livrator%TYPE);

r rec;

TYPE tab\_ind IS TABLE OF produs\_apo.nume\_produs%TYPE INDEX BY PLS\_INTEGER;

t tab\_ind;

BEGIN

select c.id\_comanda, r.nume\_rest, cl.nume\_client, cl.prenume\_client, l.nume\_livrator, l.prenume\_livrator

into r

from restaurant\_apo r join comanda\_apo c on r.id\_restaurant = c.id\_restaurant join livrator\_apo l on c.id\_livrator = l.id\_livrator join client\_apo cl on c.id\_client = cl.id\_client

where id\_comanda = idd;

DBMS\_OUTPUT.PUT\_LINE('Comanda cu id-ul ' || r.id\_com || ' a fost plasata de ' || r.nume\_cl ||' '||r.prenume\_cl||' la '||r.nume\_rest||' si livrata de '||r.nume\_liv||' '||r.prenume\_liv||' si contine urmatoarele produse: ');

select p.nume\_produs

bulk collect into t

from produs\_apo p join produs\_comanda\_apo pc on p.id\_produs = pc.id\_produs join comanda\_apo c on c.id\_comanda = pc.id\_comanda

where c.id\_comanda = idd;

for i in t.first()..t.last() loop

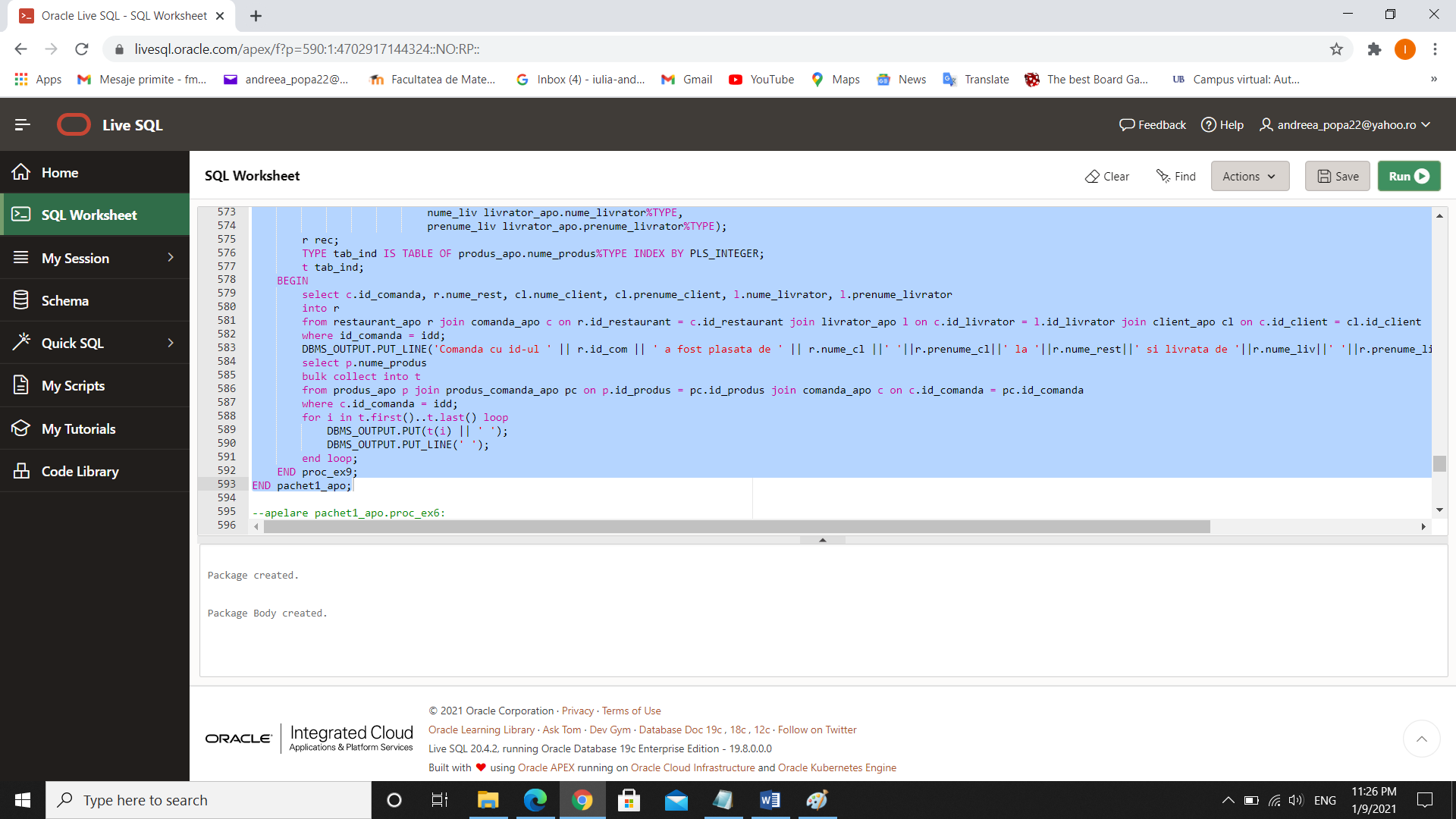
DBMS\_OUTPUT.PUT(t(i) || ' ');

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

end loop;

END proc\_ex9;

END pachet1\_apo;



--apelare pachet1\_apo.proc\_ex6:

DECLARE

TYPE rec IS RECORD (id1 client\_apo.id\_client%TYPE,

nume client\_apo.nume\_client%TYPE,

prenume client\_apo.prenume\_client%TYPE);

TYPE tab\_ind\_clienti IS TABLE OF rec INDEX BY PLS\_INTEGER;

t tab\_ind\_clienti;

age client\_apo.varsta\_client%TYPE := '19';

idd client\_apo.id\_client%TYPE := NULL;

rest restaurant\_apo.nume\_rest%TYPE;

BEGIN

SELECT distinct c.id\_client, c.nume\_client, c.prenume\_client

BULK COLLECT INTO t

FROM client\_apo c RIGHT JOIN comanda\_apo cm

ON c.id\_client = cm.id\_client

WHERE varsta\_client > age

order by c.id\_client; --punem in tabelul indexat clientii care au plasat cel putin o comanda si care au varsta mai mare decat numarul dat de la tastatura

FOR i IN t.FIRST..t.LAST LOOP

idd := t(i).id1;

pachet1\_apo.proc\_ex6(idd);

SELECT nume\_rest

INTO rest

FROM restaurant\_apo

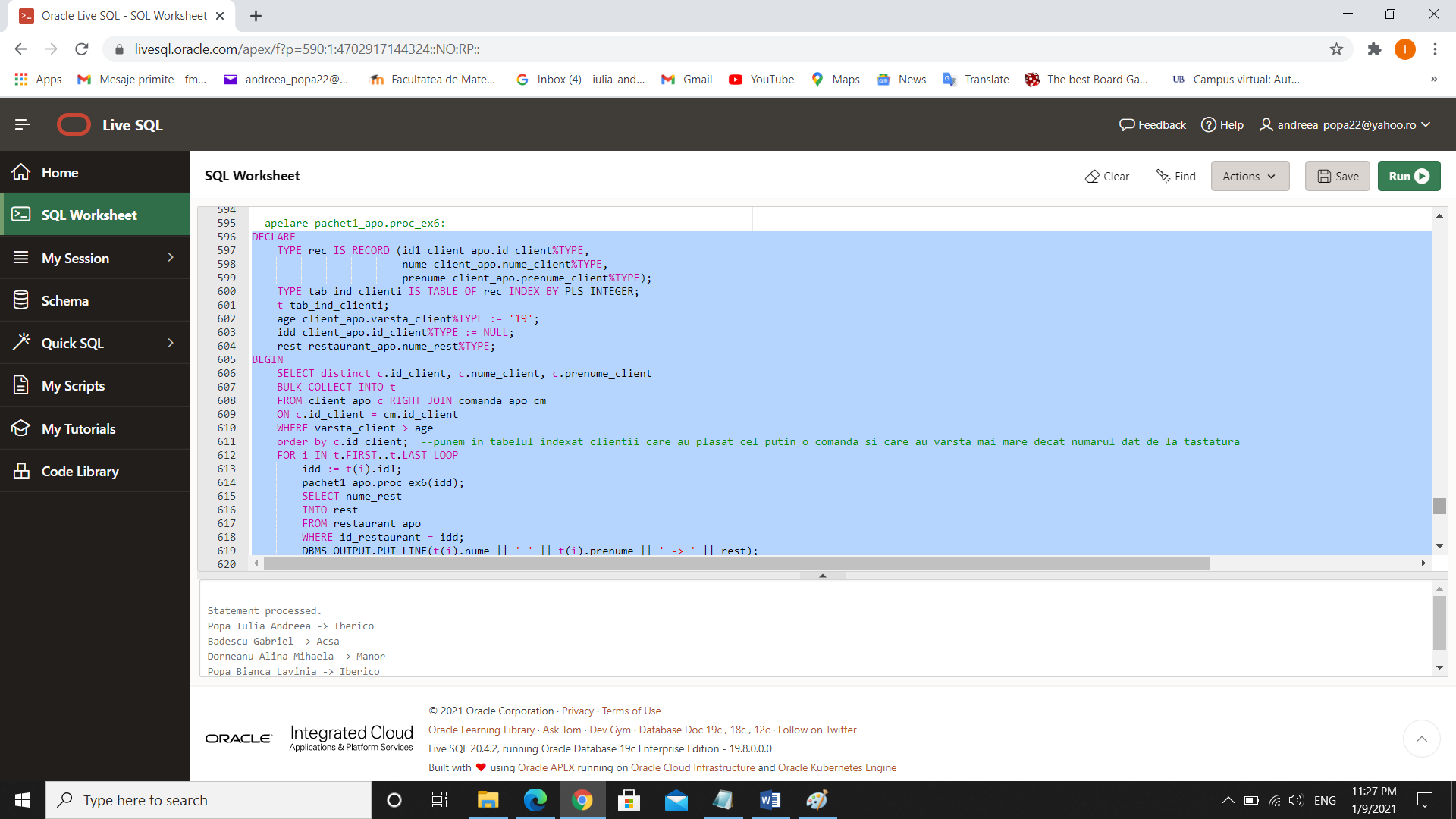
WHERE id\_restaurant = idd;

DBMS\_OUTPUT.PUT\_LINE(t(i).nume || ' ' || t(i).prenume || ' -> ' || rest);

END LOOP;

END;

/



--apelare pachet1\_apo.proc\_ex7

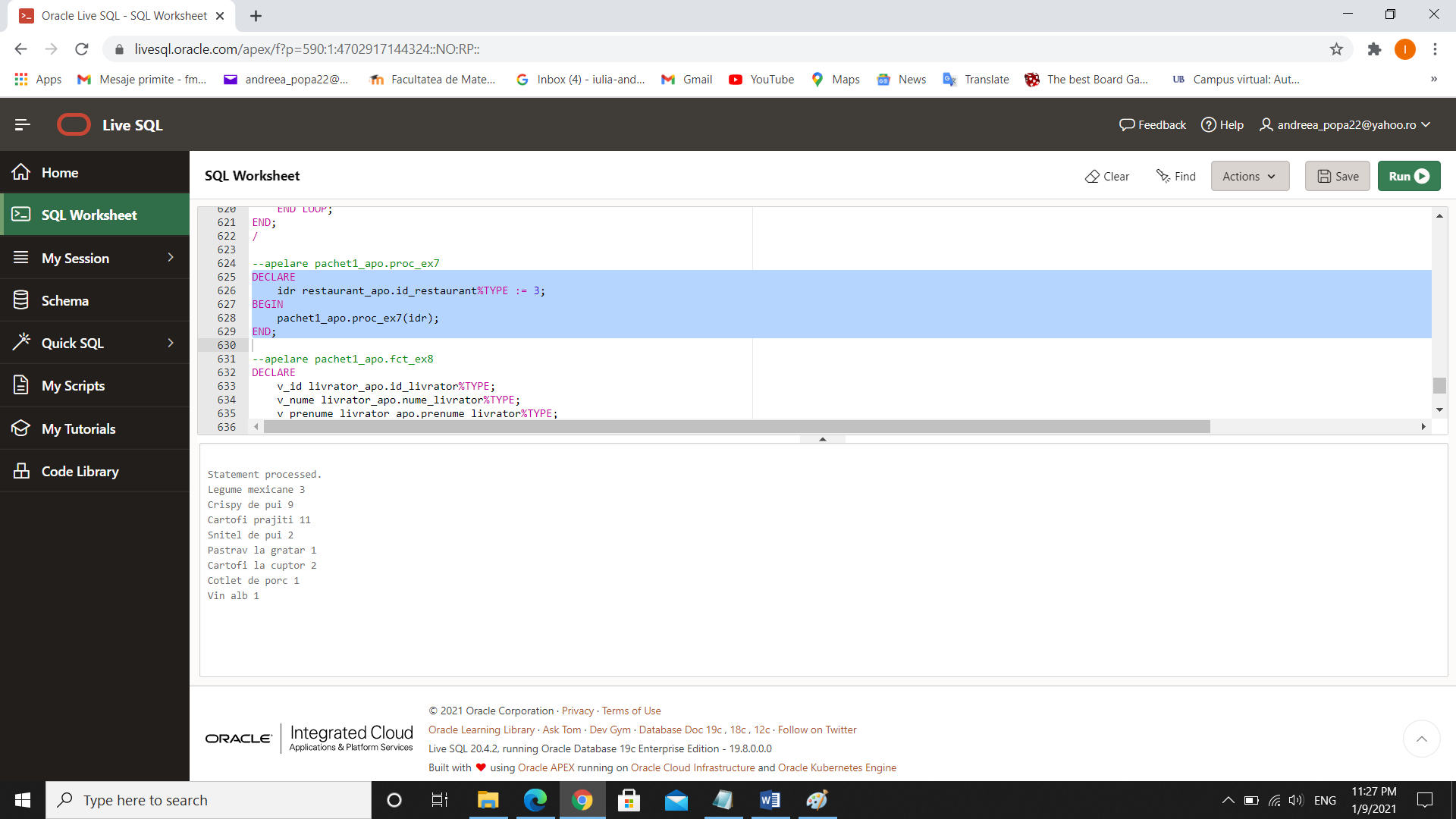
DECLARE

idr restaurant\_apo.id\_restaurant%TYPE := 3;

BEGIN

pachet1\_apo.proc\_ex7(idr);

END;



--apelare pachet1\_apo.fct\_ex8

DECLARE

v\_id livrator\_apo.id\_livrator%TYPE;

v\_nume livrator\_apo.nume\_livrator%TYPE;

v\_prenume livrator\_apo.prenume\_livrator%TYPE;

CURSOR c IS

SELECT id\_livrator, nume\_livrator, prenume\_livrator

FROM livrator\_apo;

rez VARCHAR2(150);

BEGIN

OPEN c;

LOOP

FETCH c INTO v\_id, v\_nume, v\_prenume;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT(v\_nume || ' ' || v\_prenume || ' ');

rez := pachet1\_apo.fct\_ex8(v\_id);

IF rez <> -1 THEN

DBMS\_OUTPUT.PUT\_LINE( rez || ' comenzi');

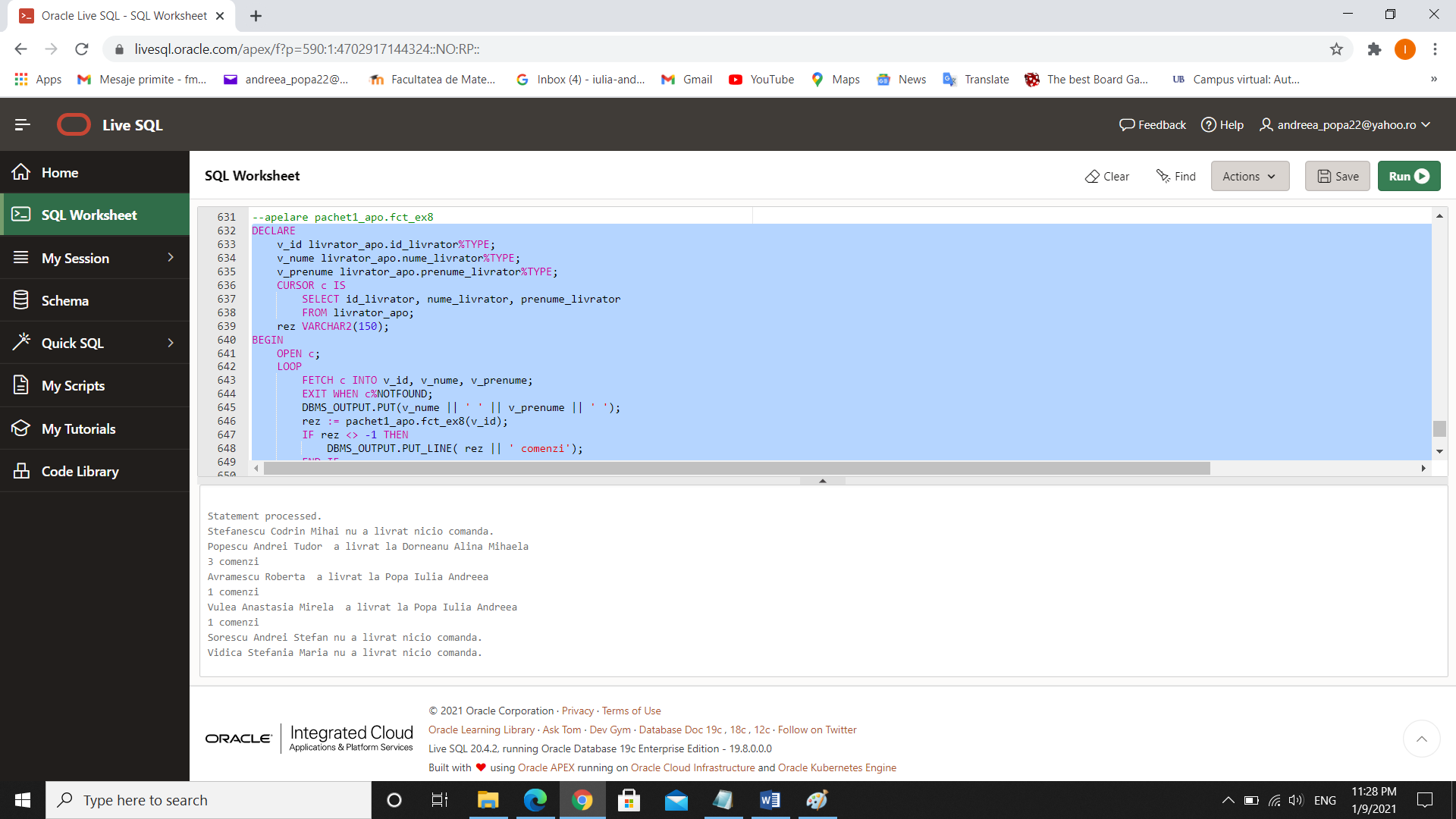
END IF;

END LOOP;

CLOSE c;

END;

/



--apelare pachet1\_apo.proc\_ex9

DECLARE

v\_data DATE := '20-DEC-02';

i number := -1;

exceptie1 exception;

BEGIN

FOR i IN (SELECT id\_comanda

from comanda\_apo

where data\_comanda = v\_data) LOOP

pachet1\_apo.proc\_ex9(i.id\_comanda);

END LOOP;

if i = -1 then raise exceptie1;

end if;

EXCEPTION

WHEN exceptie1 THEN DBMS\_OUTPUT.PUT\_LINE('Nu au fost plasate comenzi in ziua aleasa.');

END;

/

