Université de la Manouba

Rapport de Projet

Responsable Matière

Madame Hanen Brahmi

Elaboré par:

Farah Belhaj Ammar

Nour sehli

**Sommaire**

1. diagramme de classe
2. schéma relationnel

3. les scripts de création de la base de données

3.1-les scripts de création des tables

3.2-les scripts de création des utilisateurs et des privilèges

4. les requêtes d’interrogation et de modification des tables

5.les vues

6.les indexes

7.les séquences

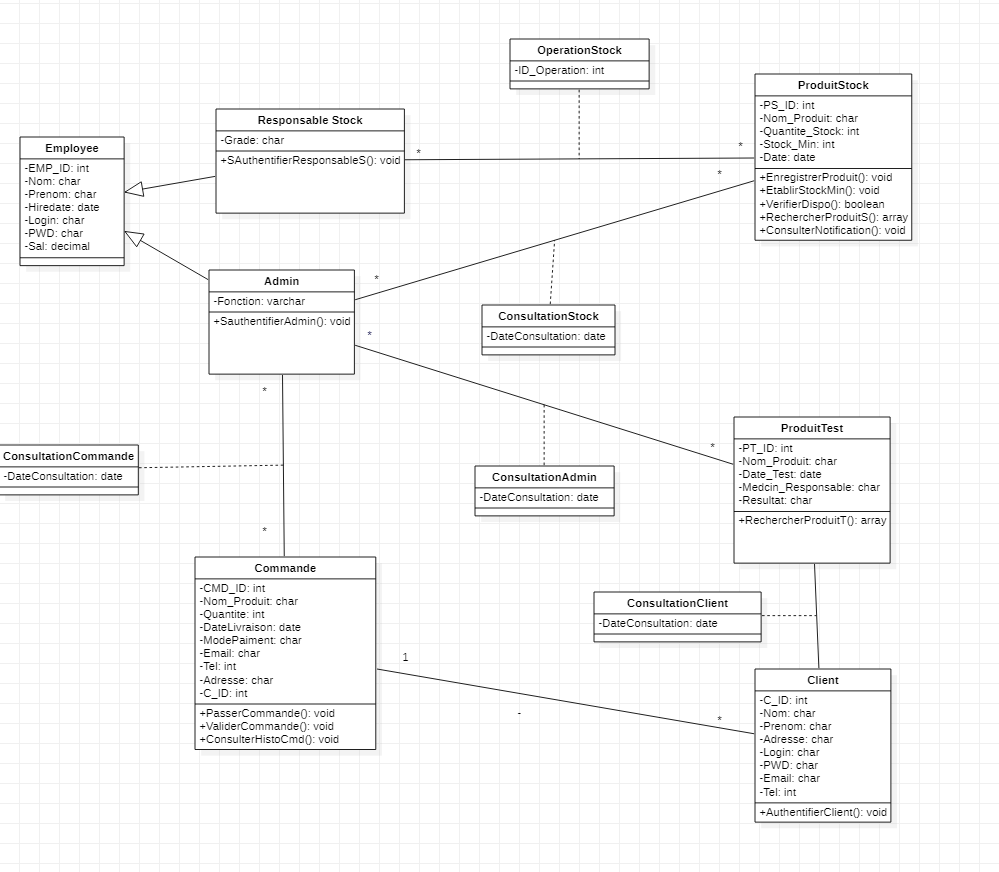
8.les procédures

9.les fonctions

10.les triggers

11.les curseurs implicites/explicites/paramètres

**1-Diagramme de classe:**



**2-le schéma relationnel:**

**Client**(C\_ID,Nom,Prenom,Adresse,Login,PWD,Email,Tel)

**Admin**(EMP\_ID,Nom,Prenom,Hiredate,Login,PWD,Sal,Fonction)

**ResponsableStock**(EMP\_ID,Nom,Prenom,Hiredate,Login,PWD,Sal,Grade)

**Commande**(CMD\_ID,Nom\_Produit,Quantité,DateLivraison,ModePaiement,Email,Tel,Adresse,#C\_ID)

**ProduitStock(**PS\_ID , Nom\_Produit,Quantite\_Stock,Stock\_Min,Date)

**ProduitTest**(PT\_ID, Nom\_Produit,Date\_Test,Medecin\_Responsable,Resultat)

**OperationStock**(#EMP\_ID , #PS\_ID, ID\_Operation)

**ConsultationAdmin(**#EMP\_ID,#PI\_ID, Date-Consultation)

**ConsultationStock**(#EMP\_ID,#PS\_ID, Date\_Consultation)

**ConsultationClient**(#C\_ID,#PI\_ID ,Date\_Consultation)

**ConsultationCommande**(#CMD\_ID,#EMP\_ID ,DateConsultation)

**3.1-les scripts de création des tables:**

\*CREATE TABLE ADMINISTRATEUR

( ADMIN\_ID NUMBER PRIMARY KEY,

ADMIN\_FIRSTNAME CHAR(20),

ADMIN\_LASTNAME CHAR(20),

FONCTION VARCHAR2(25),

SALARY NUMBER(13,0),

ADMIN\_HIRE\_DATE DATE,

LOGIN CHAR(40),

PWD CHAR(40)

);

\*CREATE TABLE CLIENT

( CLIENT\_ID NUMBER NOT NULL PRIMARY KEY,

CLIENT\_FIRSTNAME CHAR(20),

CLIENT\_LASTNAME CHAR(20),

CLIENT\_ADRESSE CHAR(40),

EMAIL CHAR(30),

TEL NUMBER,

LOGIN CHAR(40),

PWD CHAR(40)

);

\*CREATE TABLE PRODUCT\_STOCK

( PRODUCT\_STOCK\_ID NUMBER(15,0) PRIMARY KEY,

PRODUCT\_NAME CHAR(45),

PRODUCT\_QUANTITY NUMBER(15,0),

STOCK\_MIN NUMBER(15,0),

DATE\_MISE\_A\_JOUR DATE

);

\*CREATE TABLE PRODUCT\_TEST

( PRODUCT\_TEST\_ID NUMBER(\*,0) PRIMARY KEY,

PRODUCT\_NAME CHAR(45),

DATE\_TEST DATE,

MEDCIN\_RESPONSABLE\_NAME CHAR(30),

RESULTAT CHAR(10)

);

\*CREATE TABLE RESPONSABLE\_STOCK

( RESPONSABLE\_ID NUMBER(\*,0)PRIMARY KEY,

RESPONSABLE\_FIRSTNAME CHAR(20),

RESPONSABLE\_LASTNAME CHAR(20),

SALARY NUMBER(13,0),

ADMIN\_HIRE\_DATE DATE,

LOGIN CHAR(40),

PWD CHAR(40),

GRADE CHAR(10)

);

CREATE TABLE COMMANDE

( COMMANDE\_ID NUMBER PRIMARY KEY NOT NULL ,

PRODUCT\_NAME CHAR(45),

QUANTITY NUMBER,

ADRESSE CHAR(40),

EMAIL CHAR(30),

TEL NUMBER,

PAIMENT\_MODE CHAR(30),

DATE\_LIVRAISON DATE,

CLIENT\_ID NUMBER NOT NULL ENABLE,

CONSTRAINT "COMMANDE\_CON" FOREIGN KEY ("CLIENT\_ID")

REFERENCES "CLIENT" ("CLIENT\_ID")

)

**3.2 la création des utilisateurs et les privilèges :**

1-dans notre base ,on a un utilisateur admin ‘Farahbelhaj’ qui a toutes les privilèges

Et on va créer un autre utilisateur:(le responsable product\_stock)

\*Create user belhassensehli identified by belhassensehli

default tablespace product\_stock

Quota 50M on product\_stock

password expire

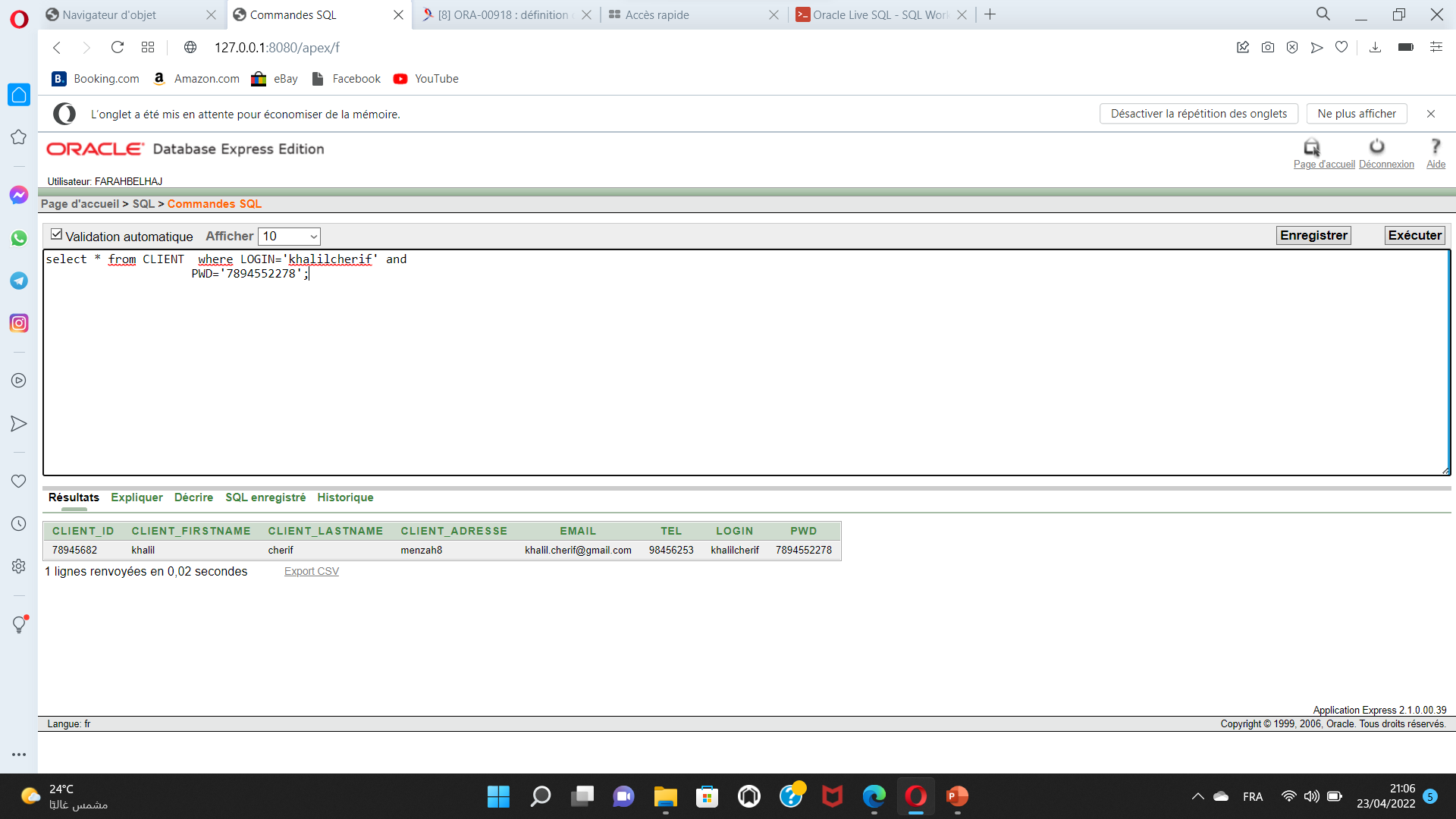
Account Lock;

\*GRANT insert,update,delete,select ON TABLE PRODUCT\_STOCK TO belhassensehli;

**4-les requêtes de modification et d’interrogation des tables:**

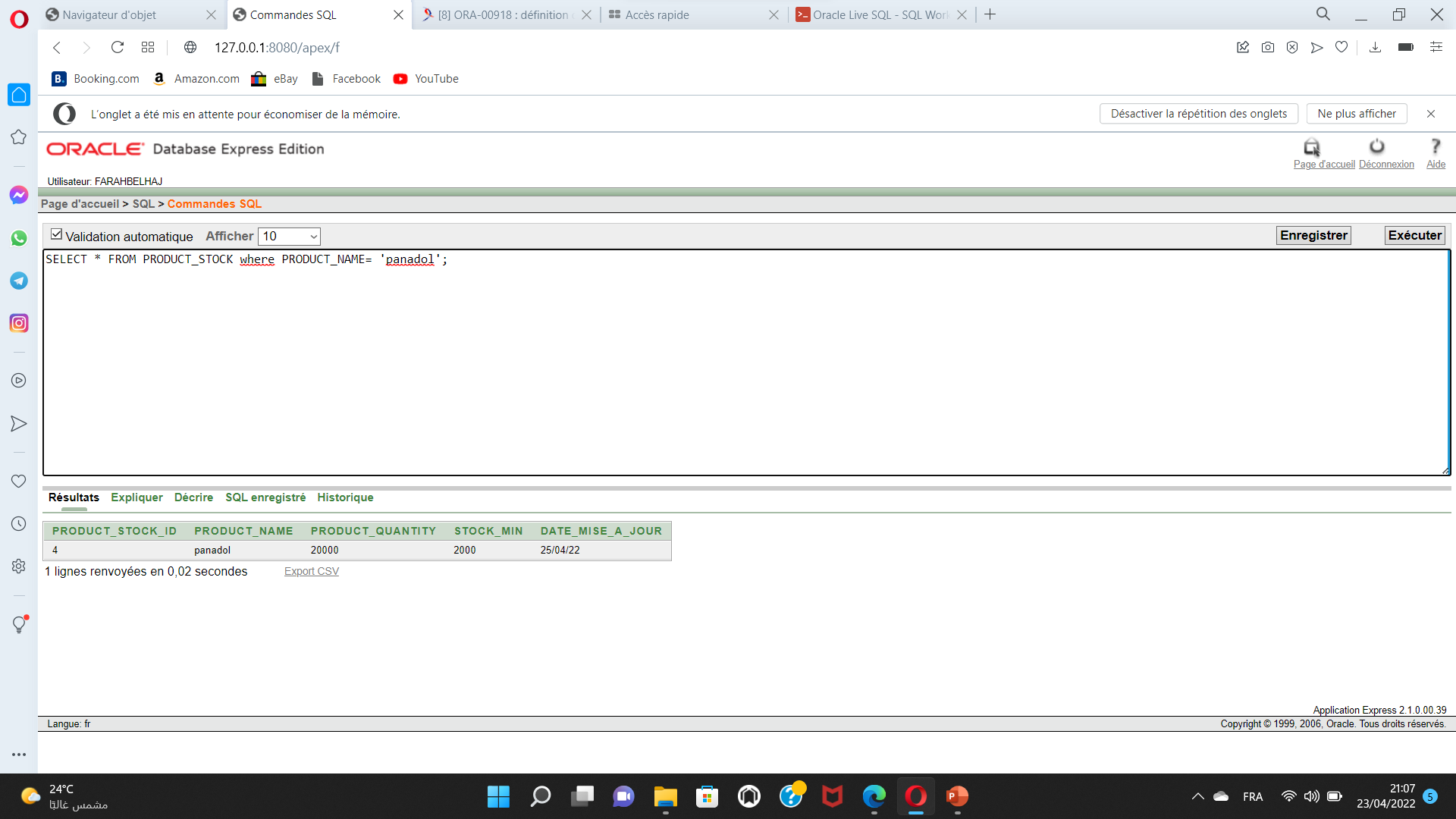
**a-requête de vérification d’authentification:**

select \* from CLIENT where LOGIN=’khalilcherif’ and PWD=’7894552278’;



**b-recherche d’un produit**

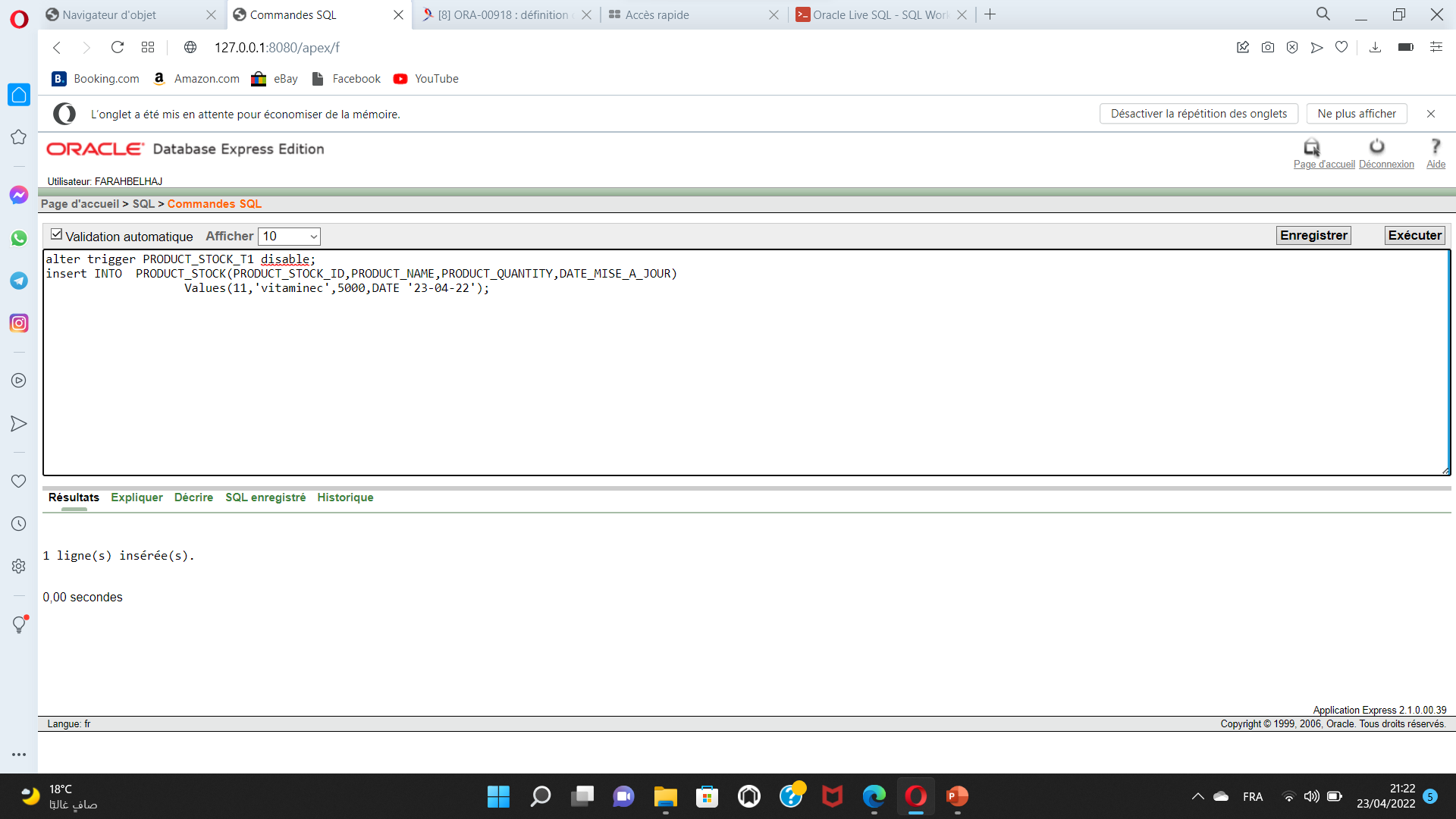
SELECT \* FROM PRODUCT\_STOCK where PRODUCT\_NAME= ‘panadol’;



**c-ajout d’un produit**

insert into PRODUCT\_STOCK(PRODUCT\_STOCK\_ID,PRODUCT\_NAME,PRODUCT\_QUANTITY,DATE\_MISE\_A\_JOUR)

Values(11,’vitaminec’,5000,DATE ‘23-04-22’);

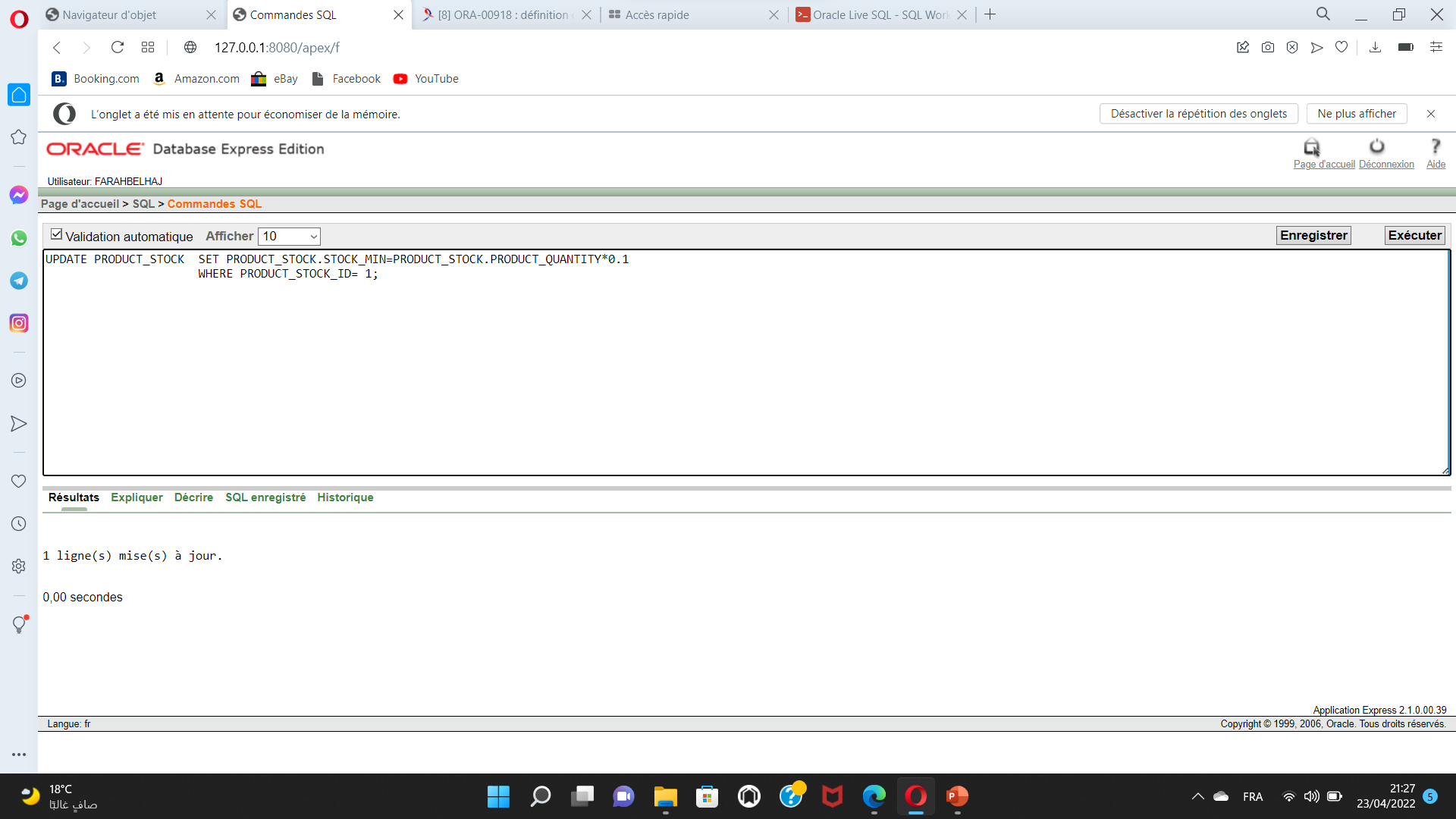


**d-établir un niveau de stock minimum**

UPDATE PRODUCT\_STOCK SET

PRODUCT\_STOCK.STOCK\_MIN=PRODUCT\_STOCK.PRODCUT\_QUANTITY\*0.1

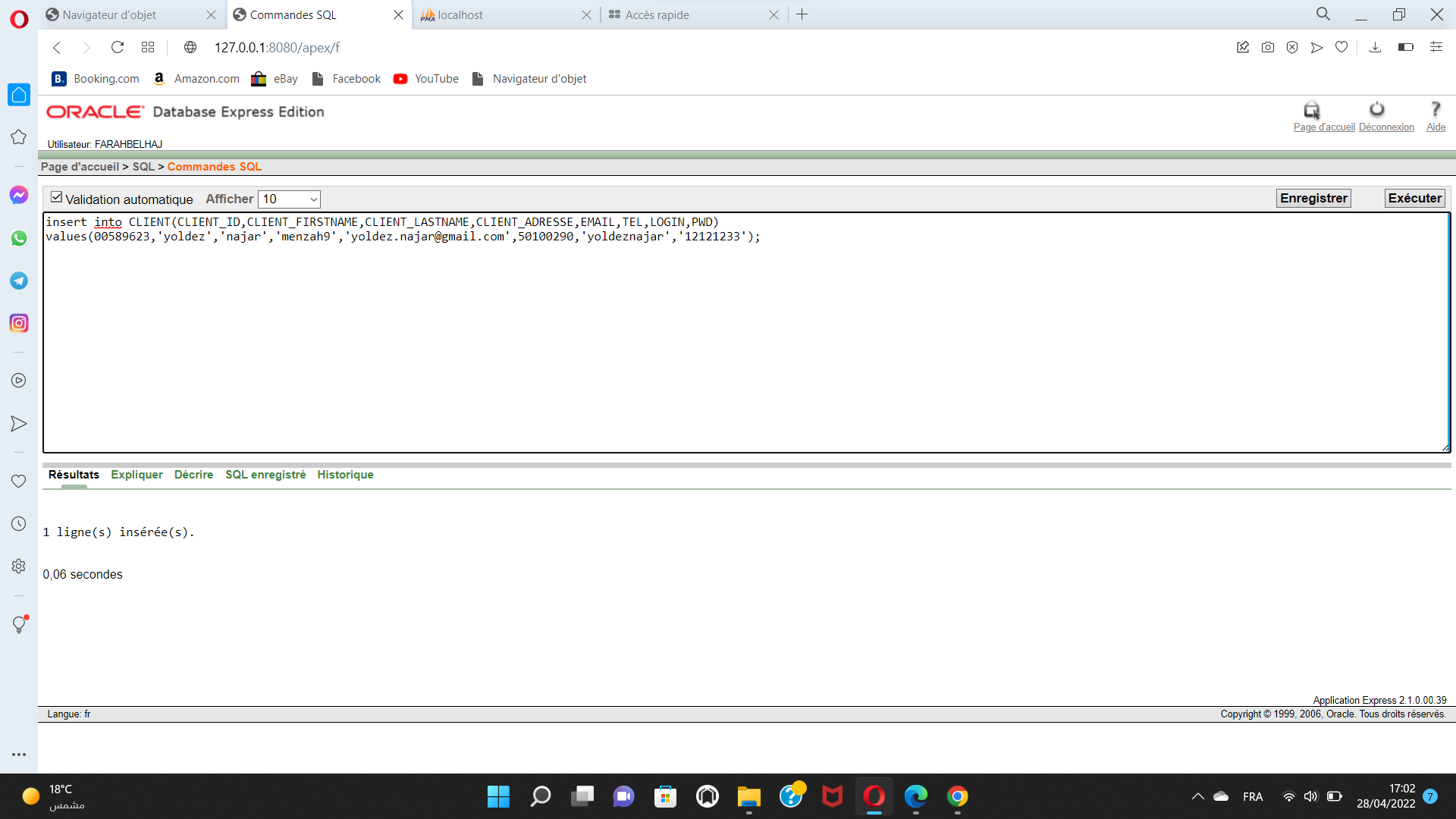
WHERE **PRODUCT\_STOCK\_ID**= 1;

****

**e-insérer un client:**

insert into CLIENT(CLIENT\_ID,CLIENT\_FIRSTNAME,CLIENT\_LASTNAME,CLIENT\_ADRESSE,EMAIL,TEL,LOGIN,PWD)

values(00589623,'yoldez','najar','menzah9','yoldez.najar@gmail.com',50100290,'yoldeznajar','12121233');

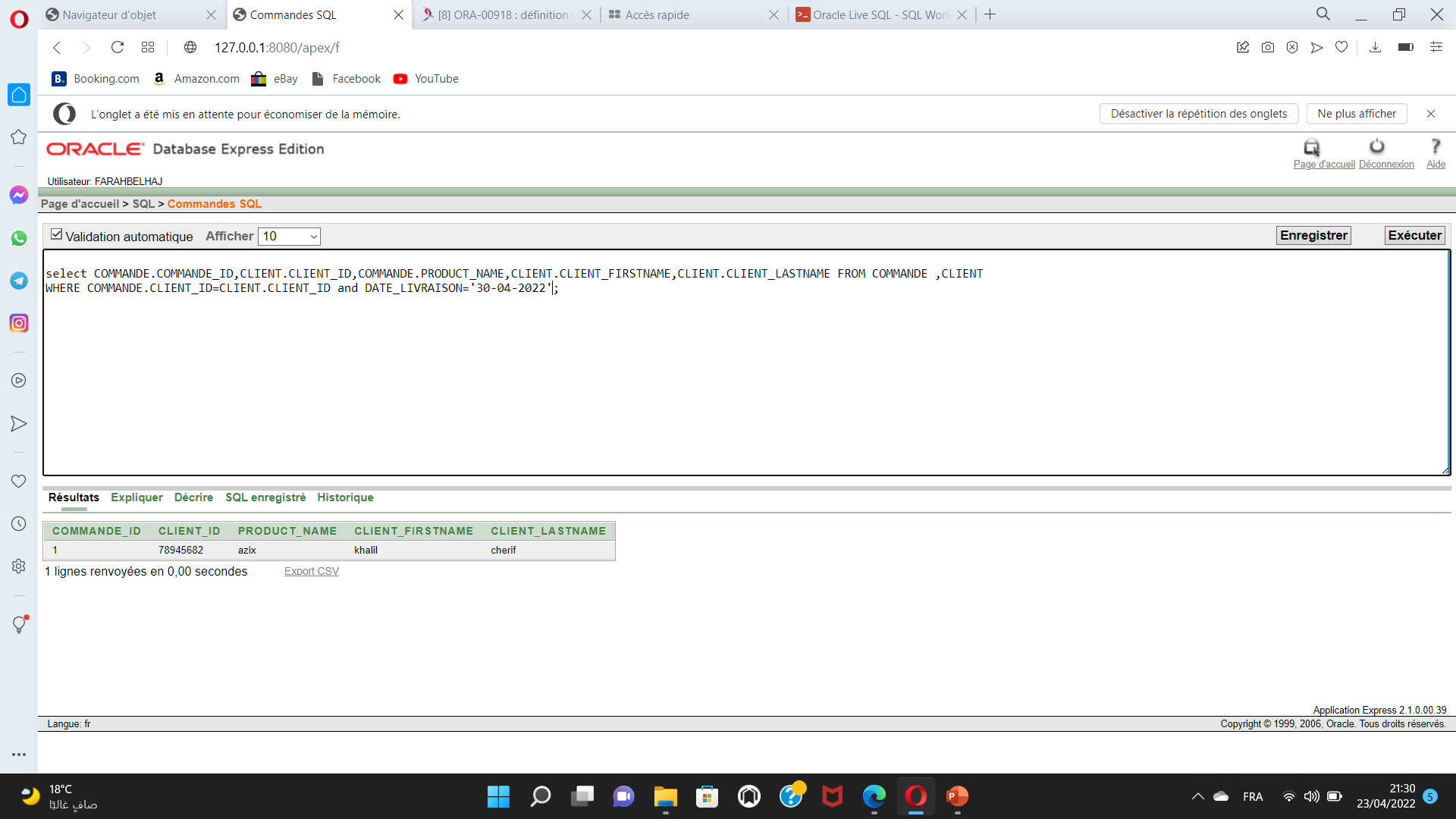


**f-consulter les livraisons à une date donné:**

select COMMANDE.COMMANDE\_ID,CLIENT.CLIENT\_ID,COMMANDE.PRODUCT\_NAME,CLIENT.CLIENT\_FIRSTNAME,CLIENT.CLIENT\_LASTNAME from COMMANDE ,CLIENT

WHERE COMMANDE.CLIENT\_ID=CLIENT.CLIENT\_ID and DATE\_LIVRAISON=’30-04–22’;

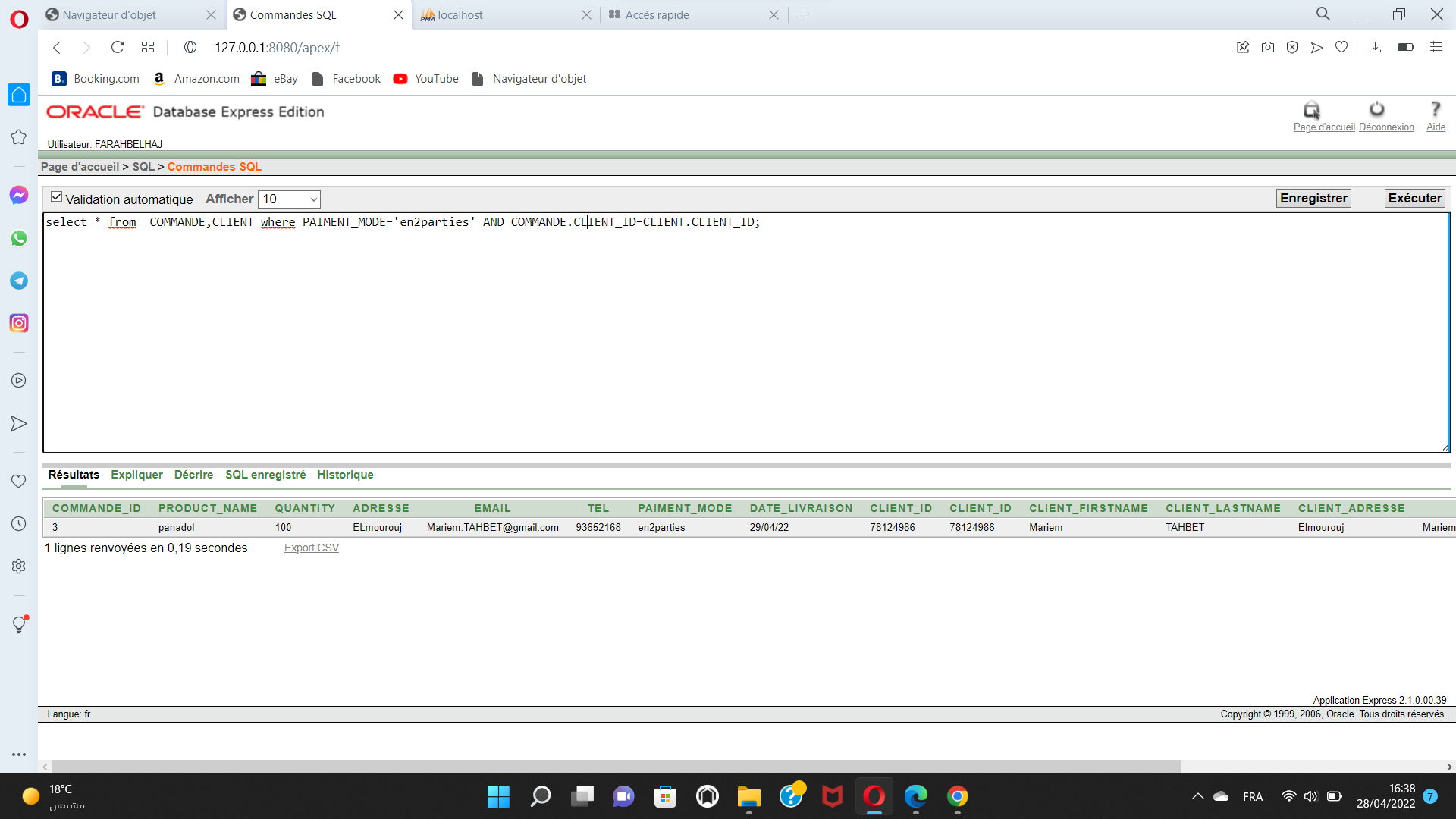
**//une jointure entre la table client et la table commande**



**g-afficher toutes les informations sur un client dont son mode de paiement=’en2parties’**

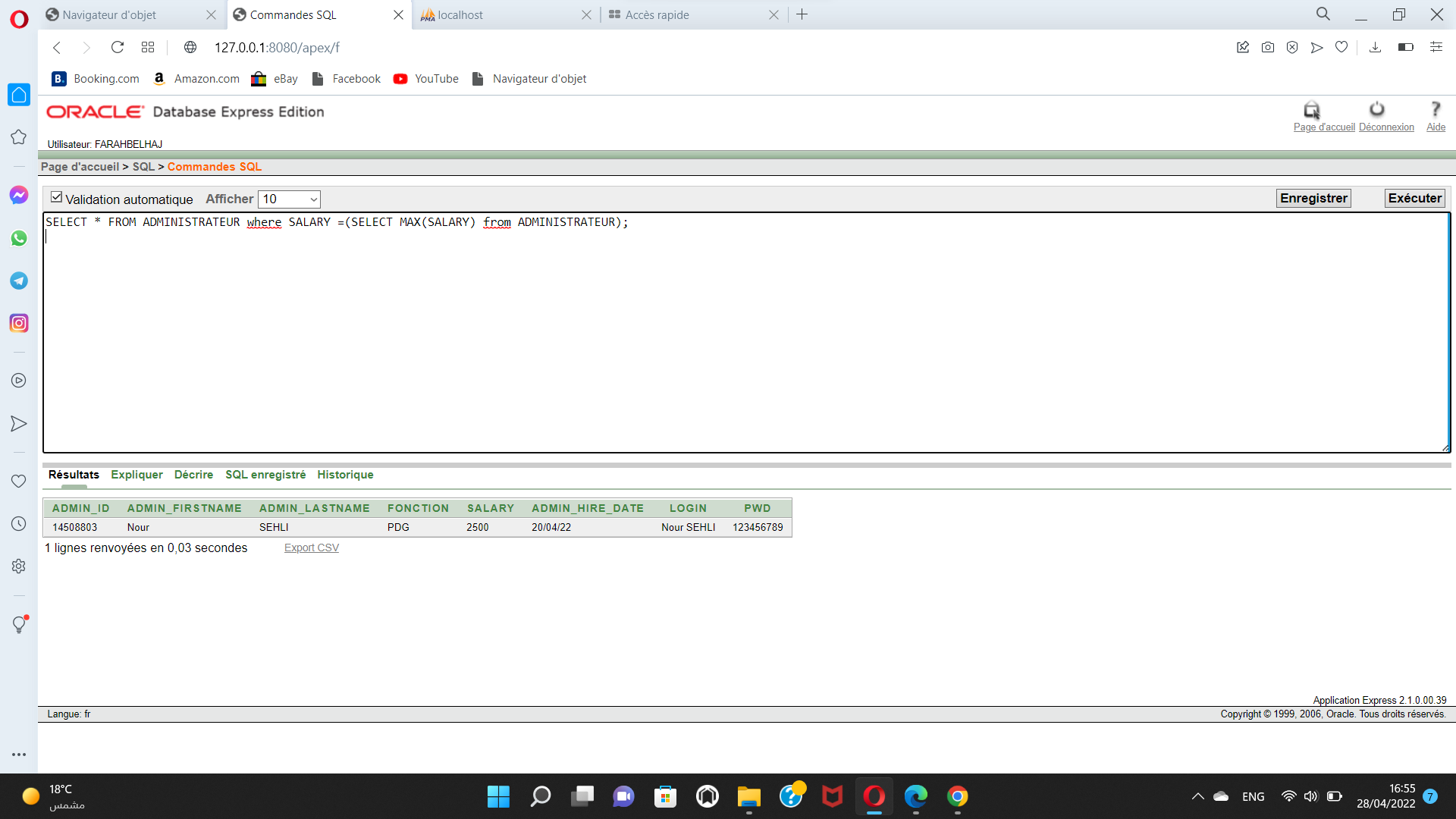
select \* from COMMANDE,CLIENT where PAIMENT\_MODE='en2parties' AND COMMANDE.CLIENT\_ID=CLIENT.CLIENT\_ID;

**//une jointure entre la table client et la table commande**



**h-afficher l’admin qui a le salaire le plus élevé:**

SELECT \* FROM ADMINISTRATEUR where SALARY =(SELECT MAX(SALARY) from ADMINISTRATEUR);



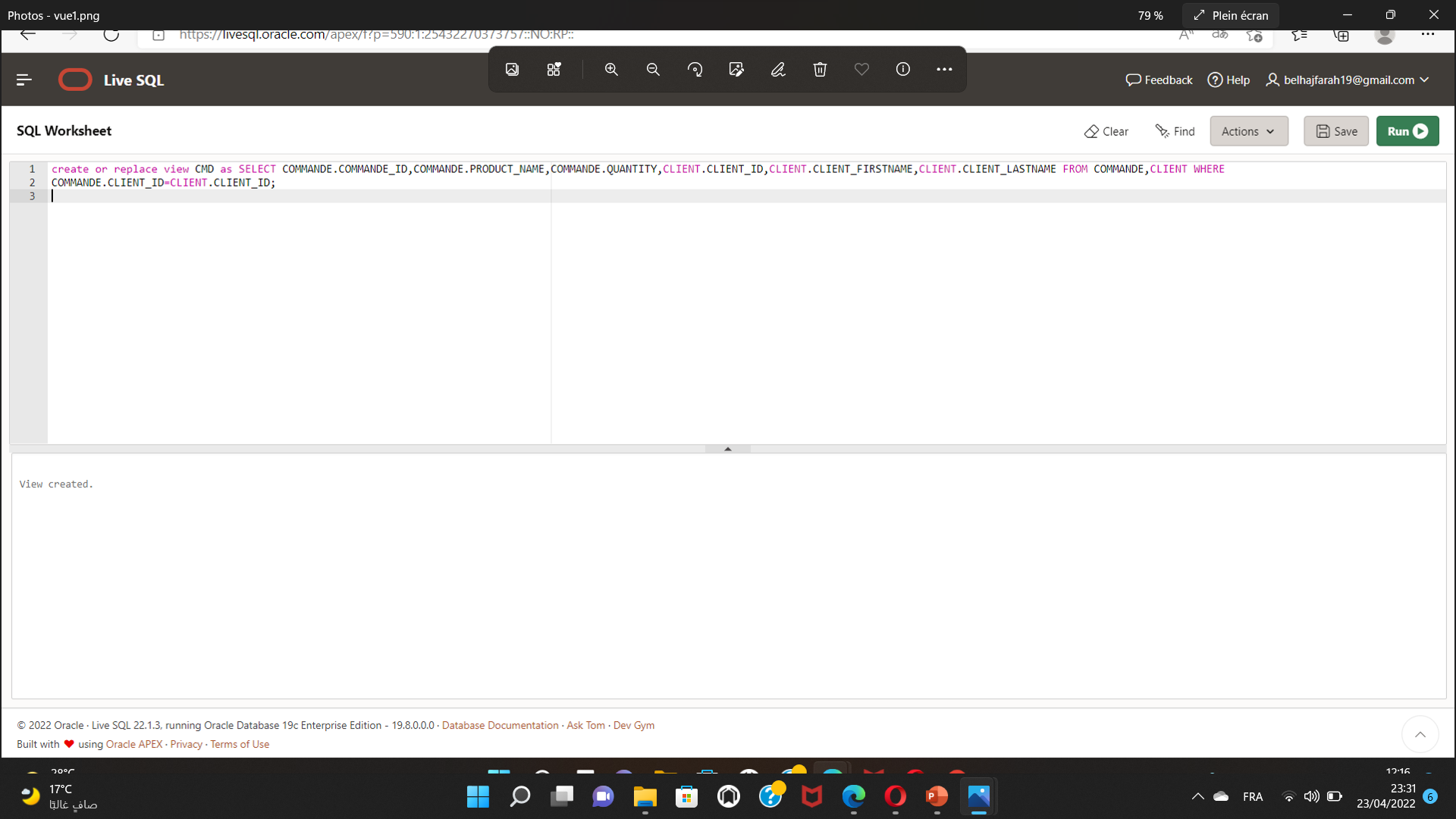
**5-les vues:**

a /création d’une vue qui rassemble les données les plus importants d’une commande avec les données d’un client

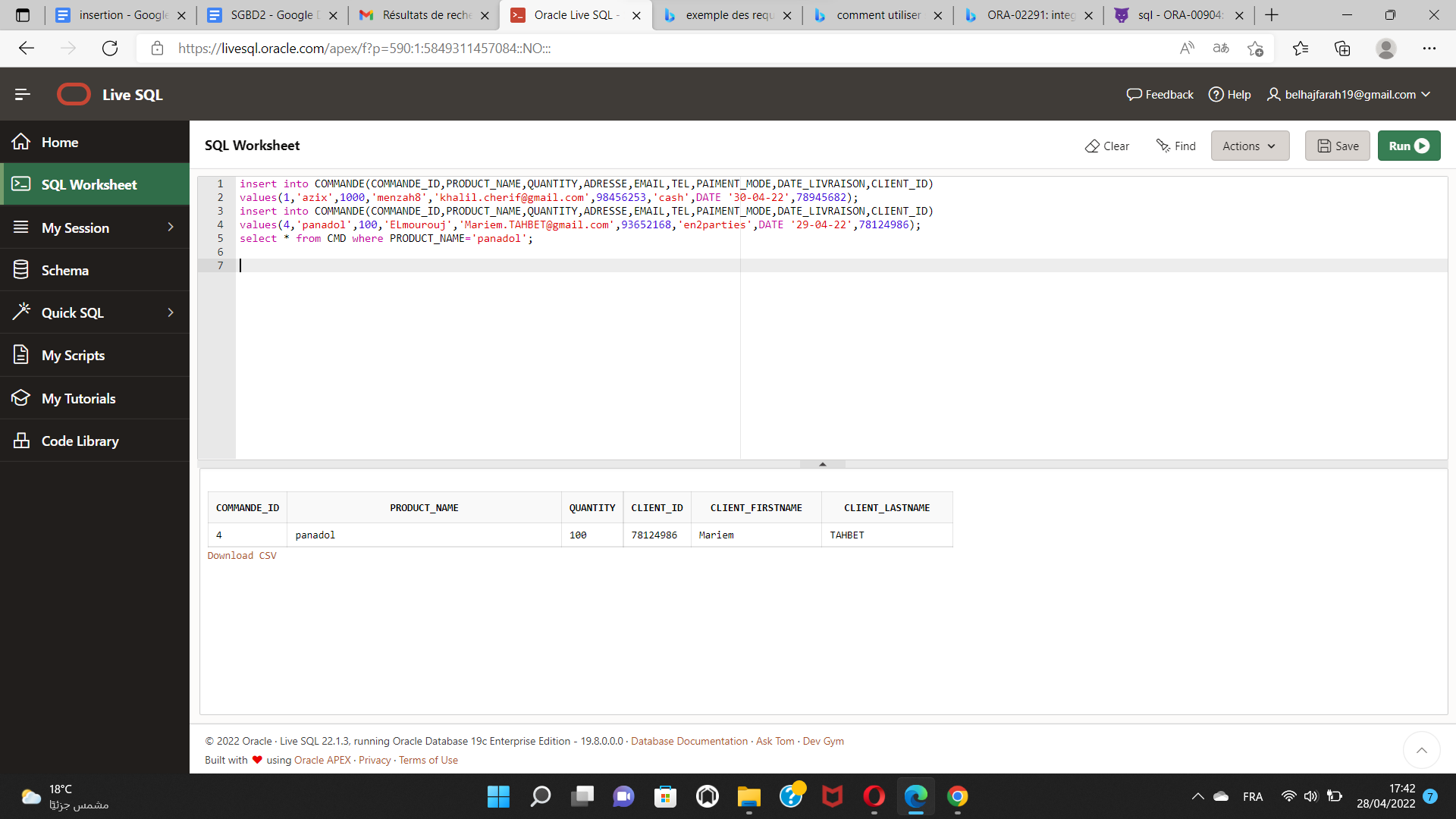
create or replace view CMD as SELECT COMMANDE.COMMANDE\_ID,COMMANDE.PRODUCT\_NAME,COMMANDE.QUANTITY,CLIENT.CLIENT\_ID,CLIENT.CLIENT\_FIRSTNAME,CLIENT.CLIENT\_LASTNAME FROM COMMANDE,CLIENT WHERE

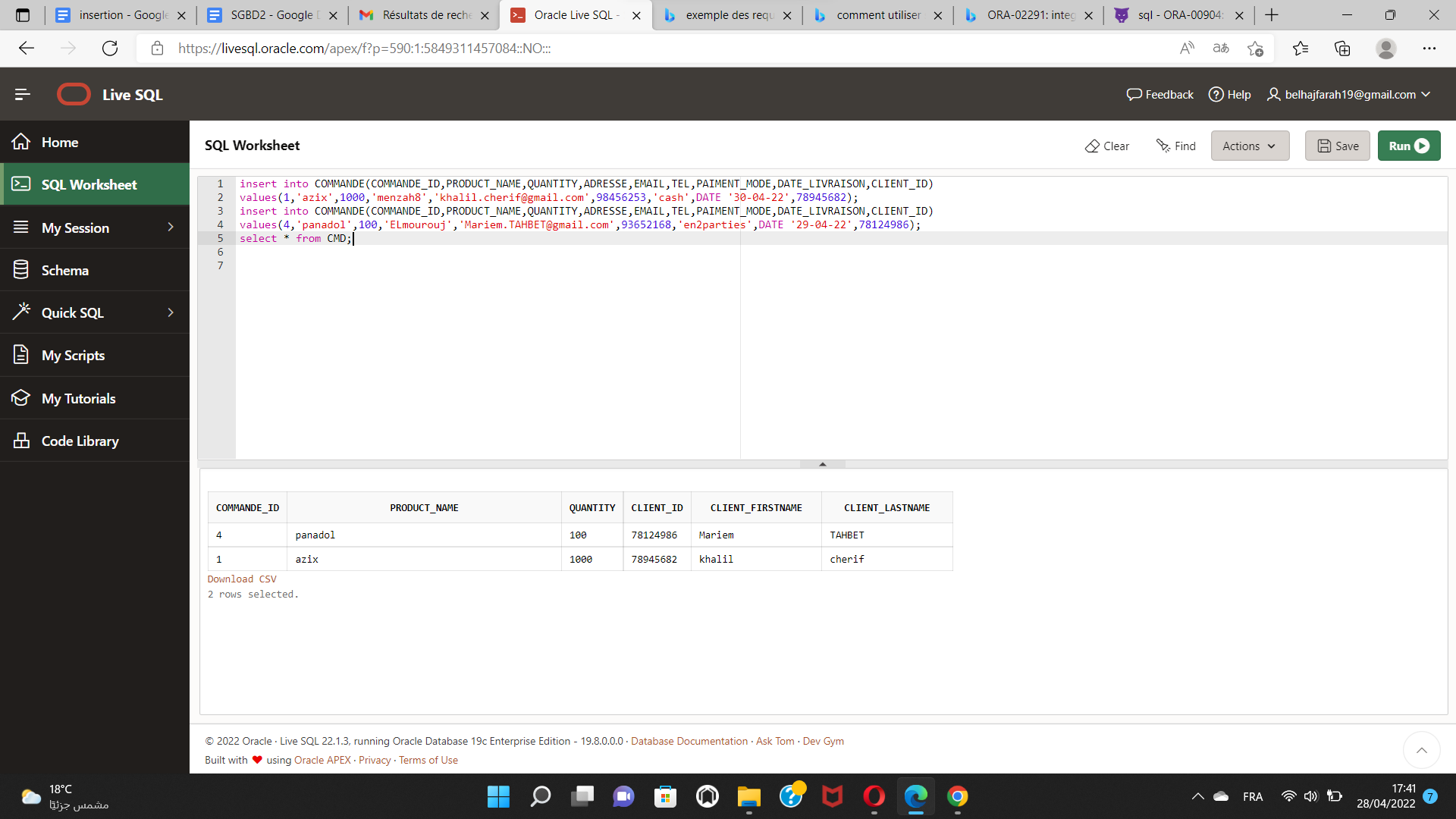
COMMANDE.CLIENT\_ID=CLIENT.CLIENT\_ID;

**//jointure de table commande avec la table client**



***\*manipulation des vues:***



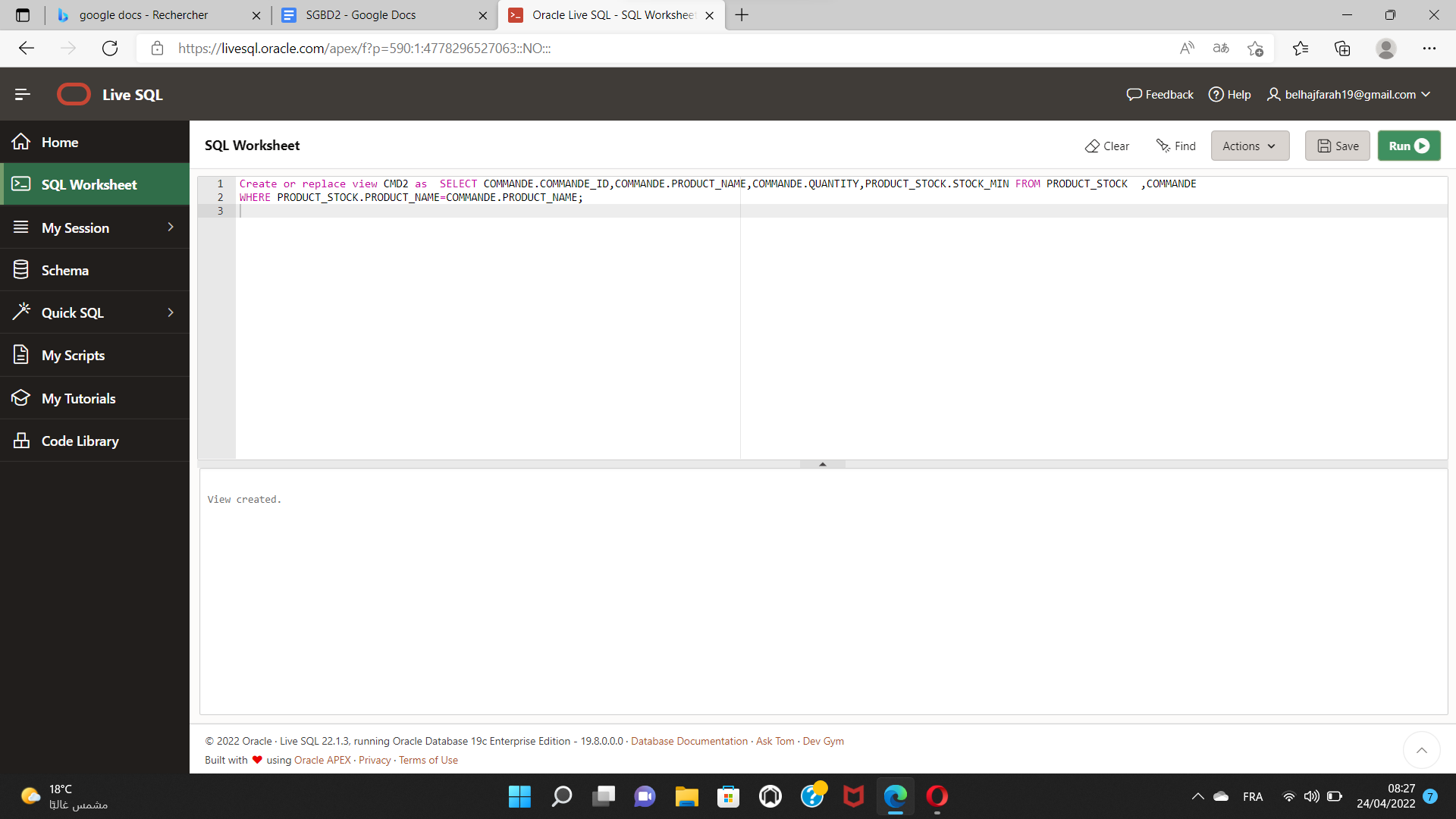


b-création d’une vue qui intègre la quantité de produit commandé et la quantité de stock disponible

Create or replace view CMD2 as SELECT COMMANDE.COMMANDE\_ID,COMMANDE.PRODUCT\_NAME,COMMANDE.QUANTITY,PRODUCT\_STOCK.STOCK\_MIN FROM PRODUCT\_STOCK ,COMMANDE

WHERE PRODUCT\_STOCK.PRODUCT\_NAME=COMMANDE.PRODUCT\_NAME;

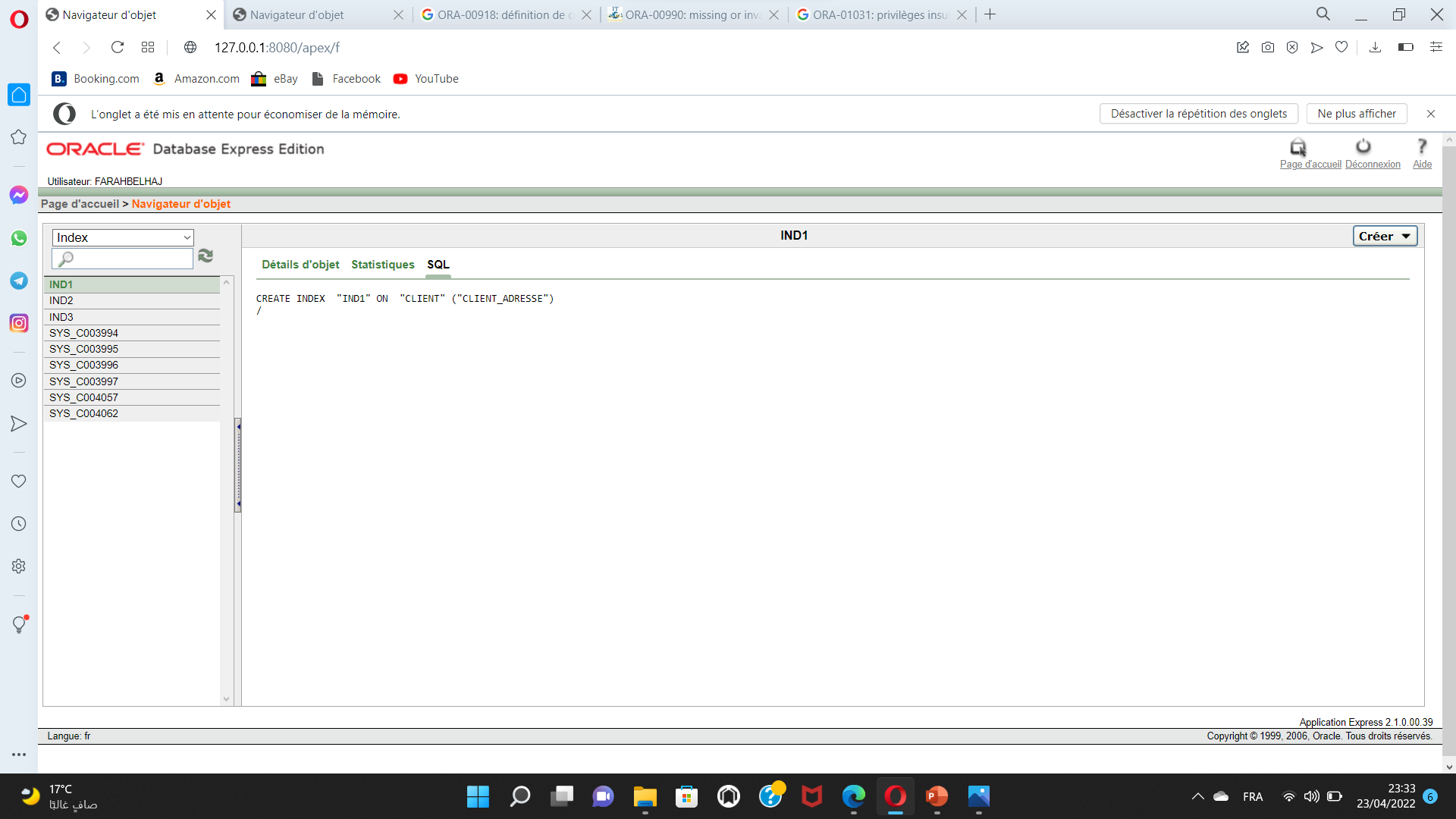
**//jointure entre la table commande et la table product\_stock**

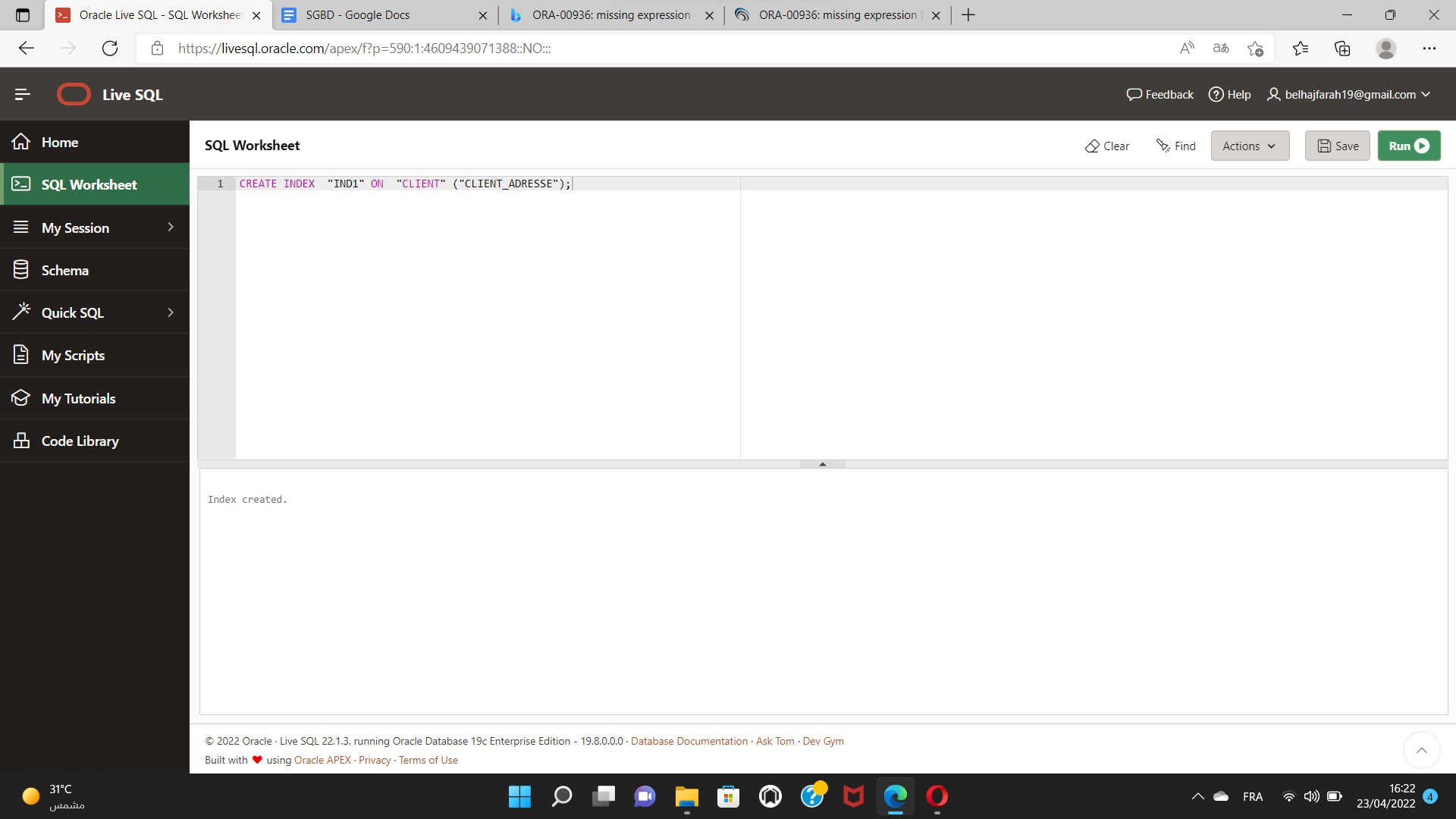


**6-les indexes:**

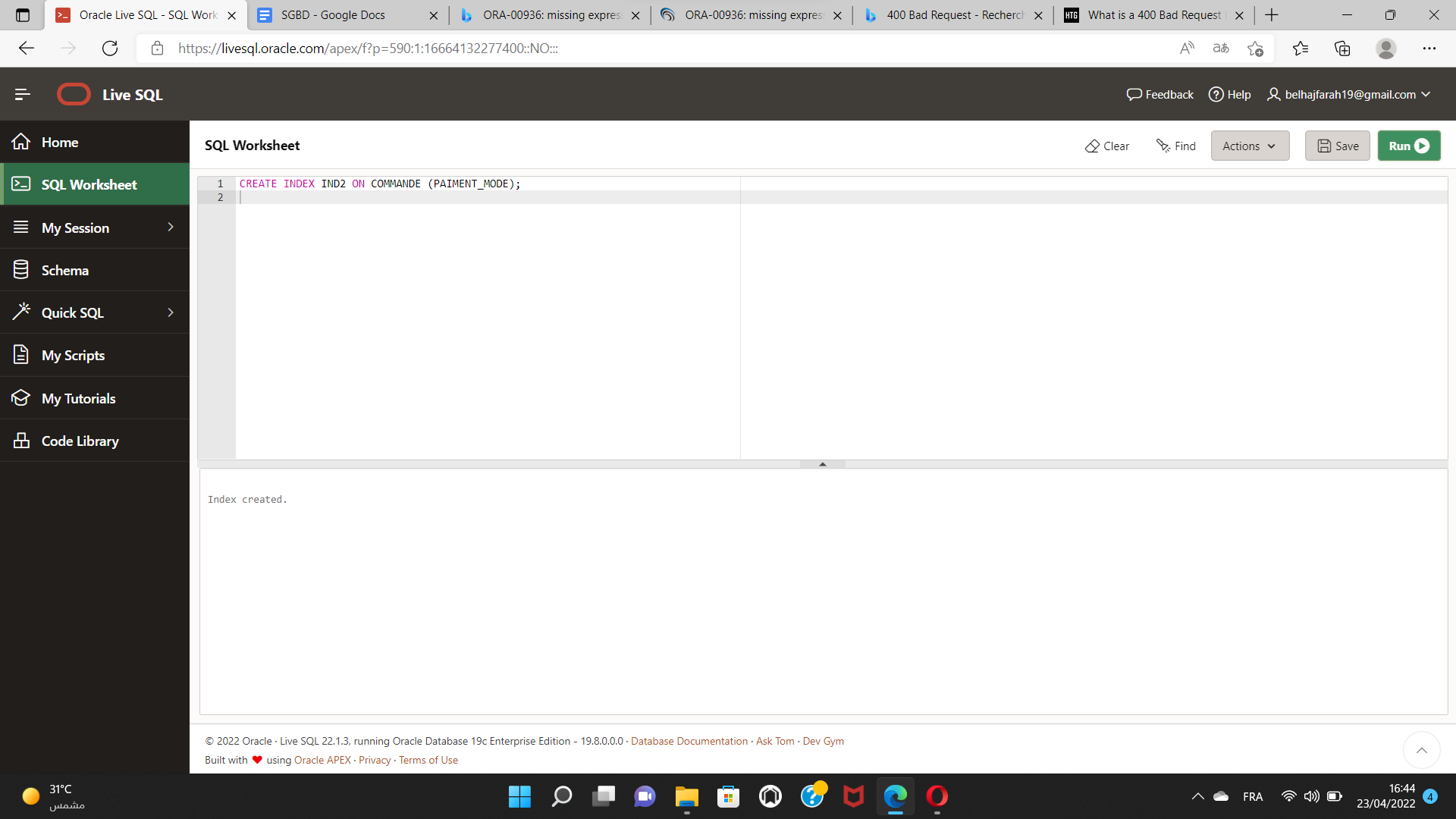
\*CREATE INDEX "IND1" ON "CLIENT" ("CLIENT\_ADRESSE");

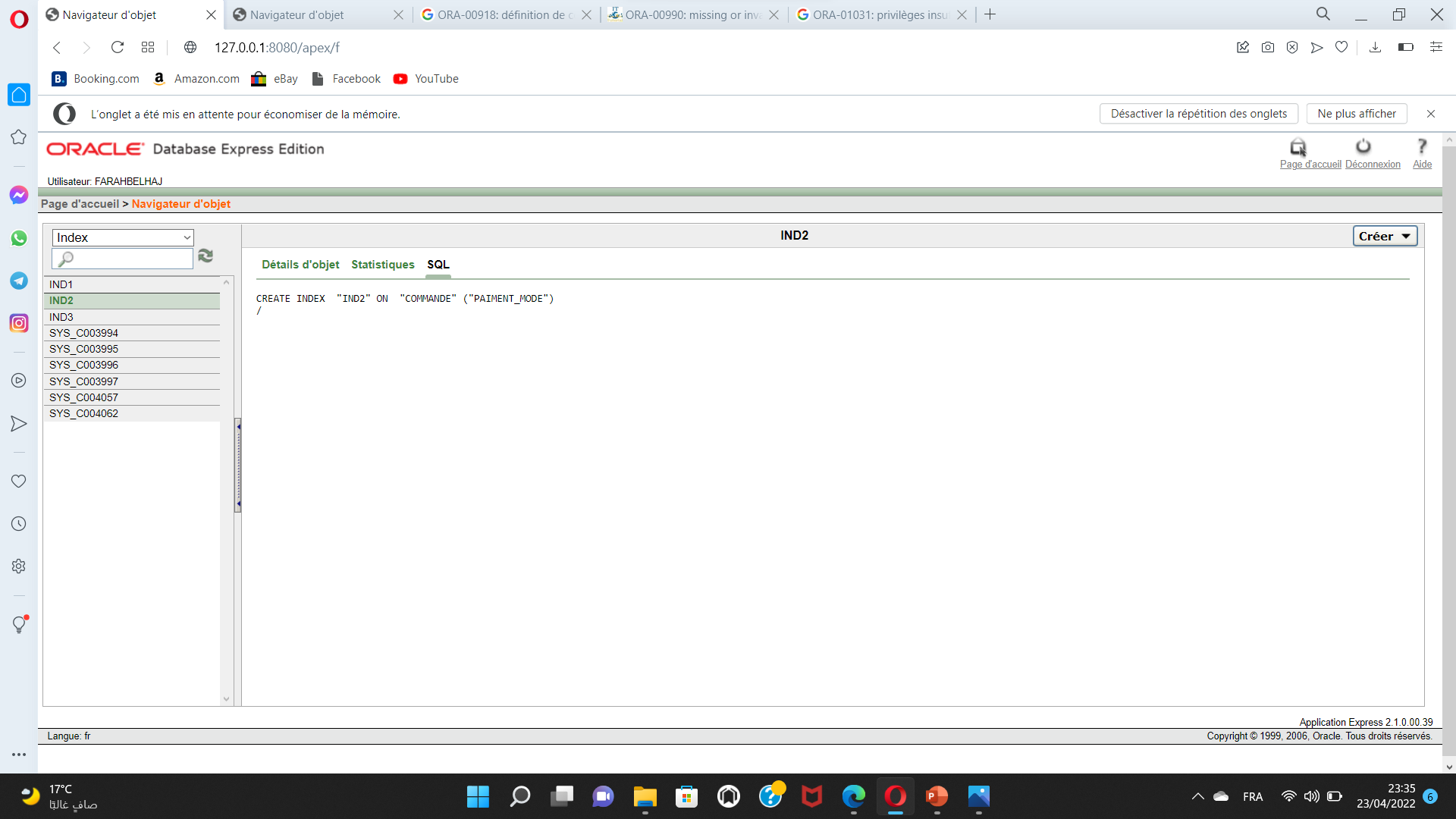
**//vu que il ya beaucoup d’adresses qui se répètent**



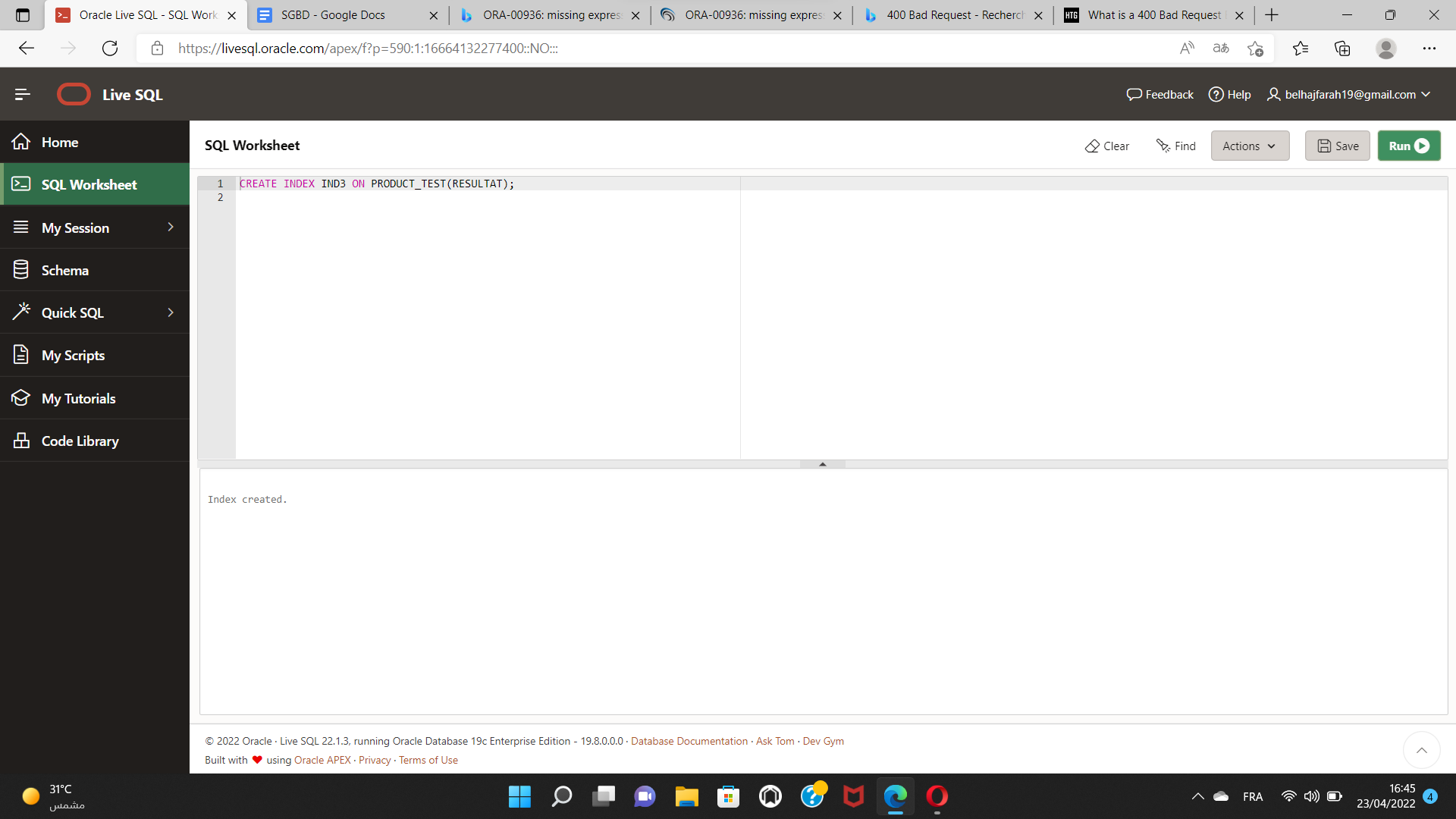


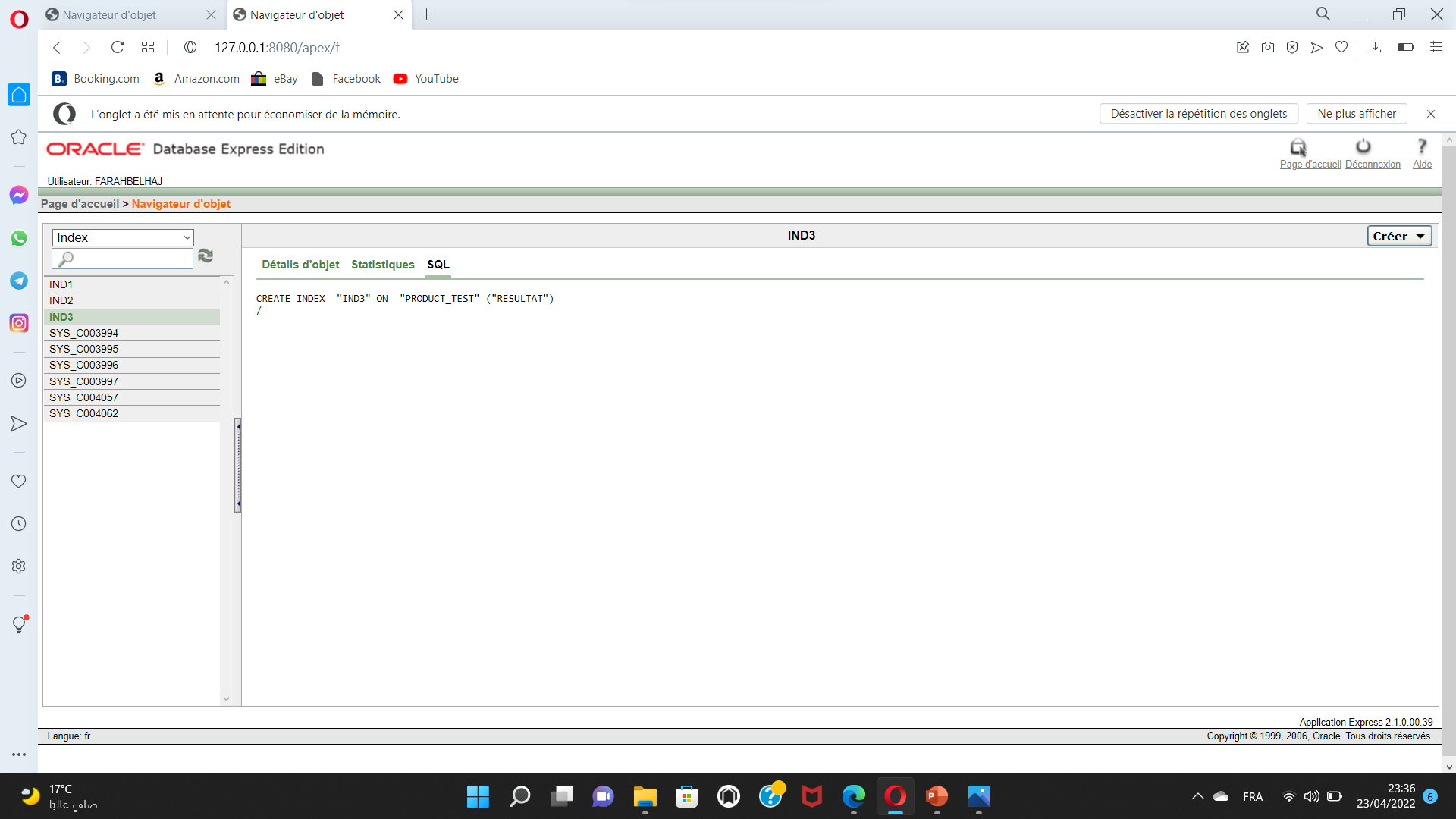
\*CREATE INDEX IND2 ON COMMANDE (PAIMENT\_MODE);





\*CREATE INDEX IND3 ON PRODUCT\_TEST(RESULTAT);





**7-les séquences:**

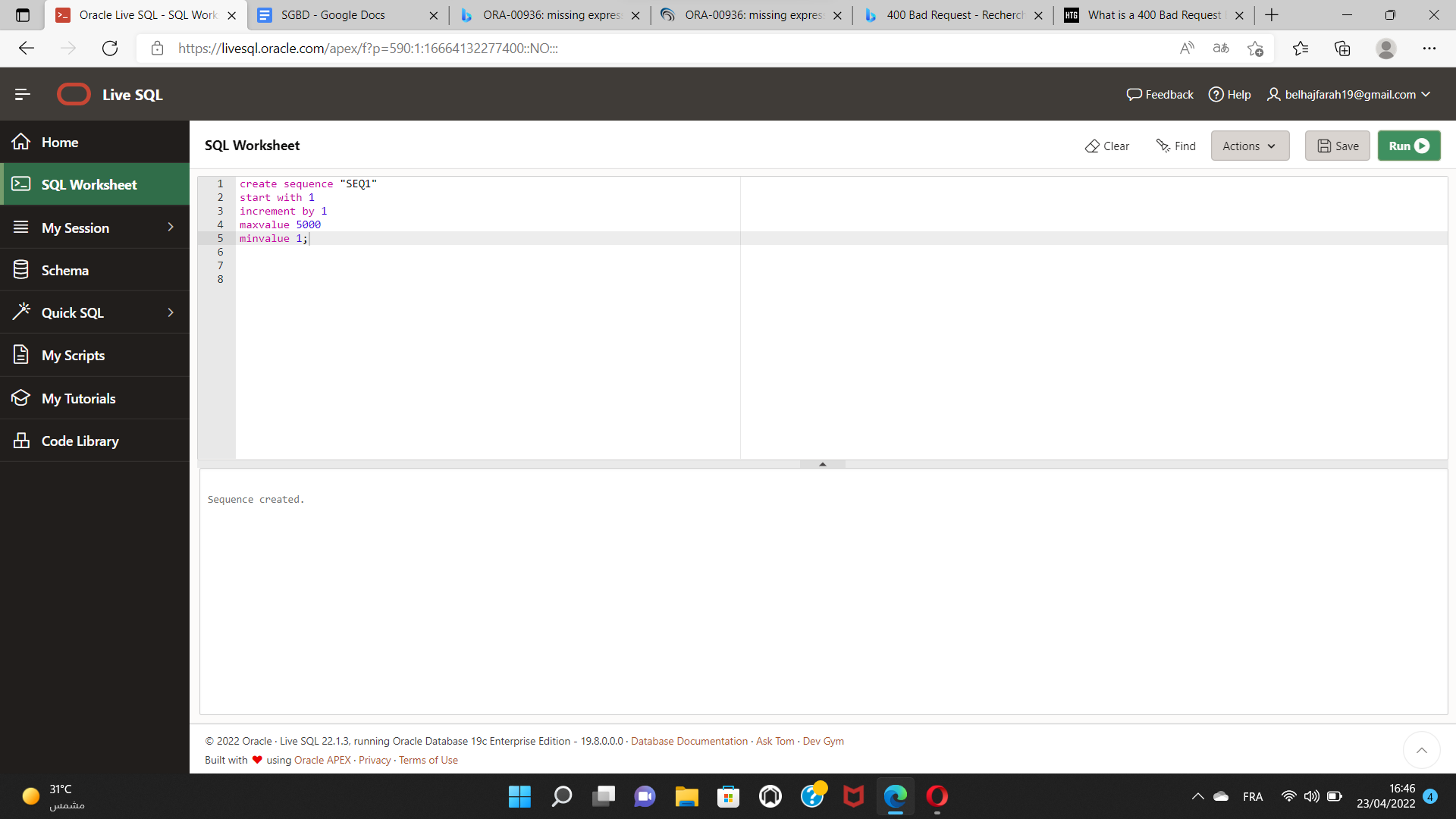
create sequence "SEQ1"

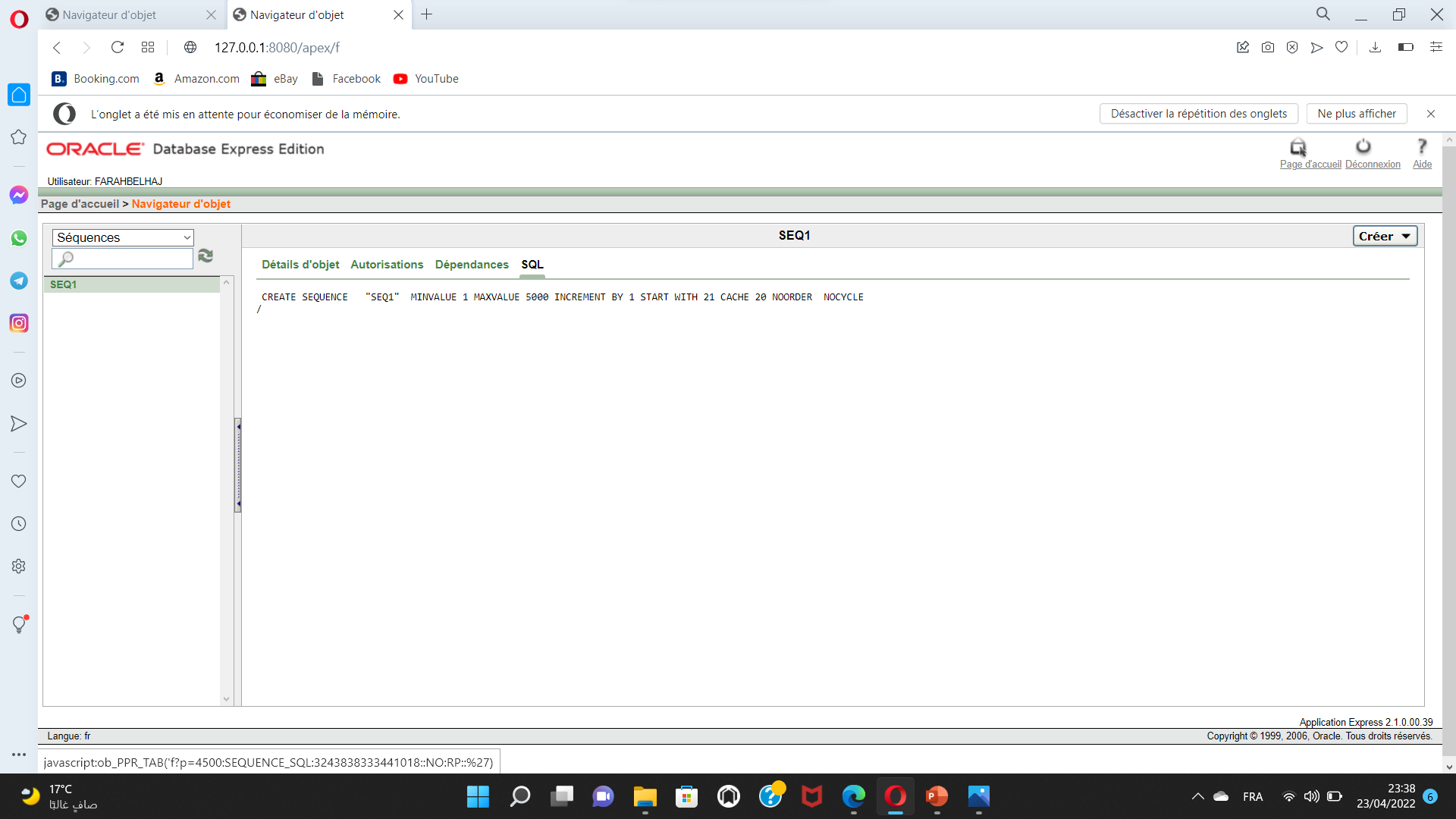
start with 1

increment by 1

maxvalue 5000

minvalue 1





**8-les procédures:**

**QUESTION: écrire une procédure qui retourne la quantité de stock courant et le stock min d’un médicament donnée**

CREATE OR REPLACE PROCEDURE STOCK\_PR

(p\_nom IN varchar2,p\_num1 OUT NUMBER,p\_num2 OUT NUMBER )

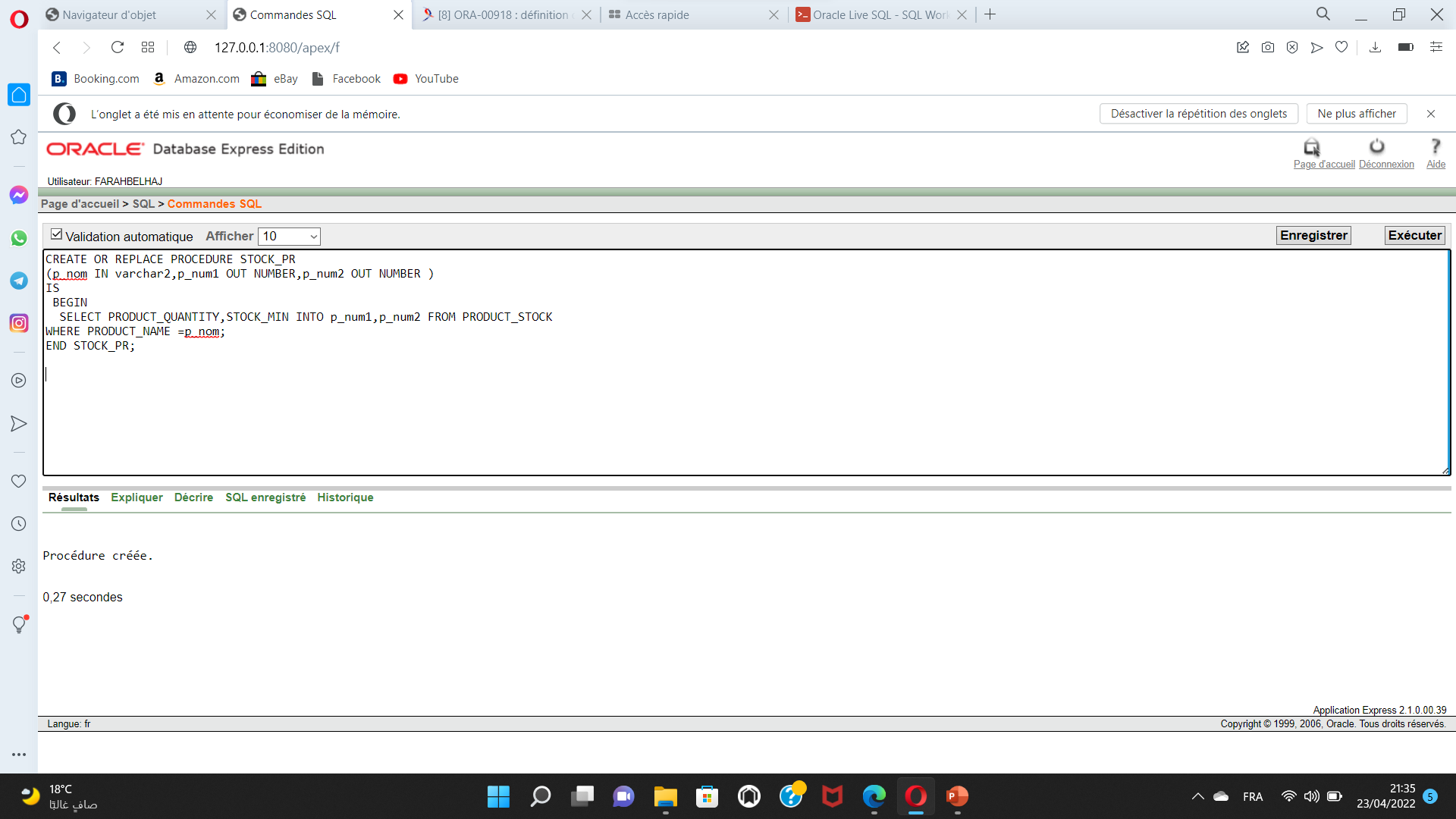
IS

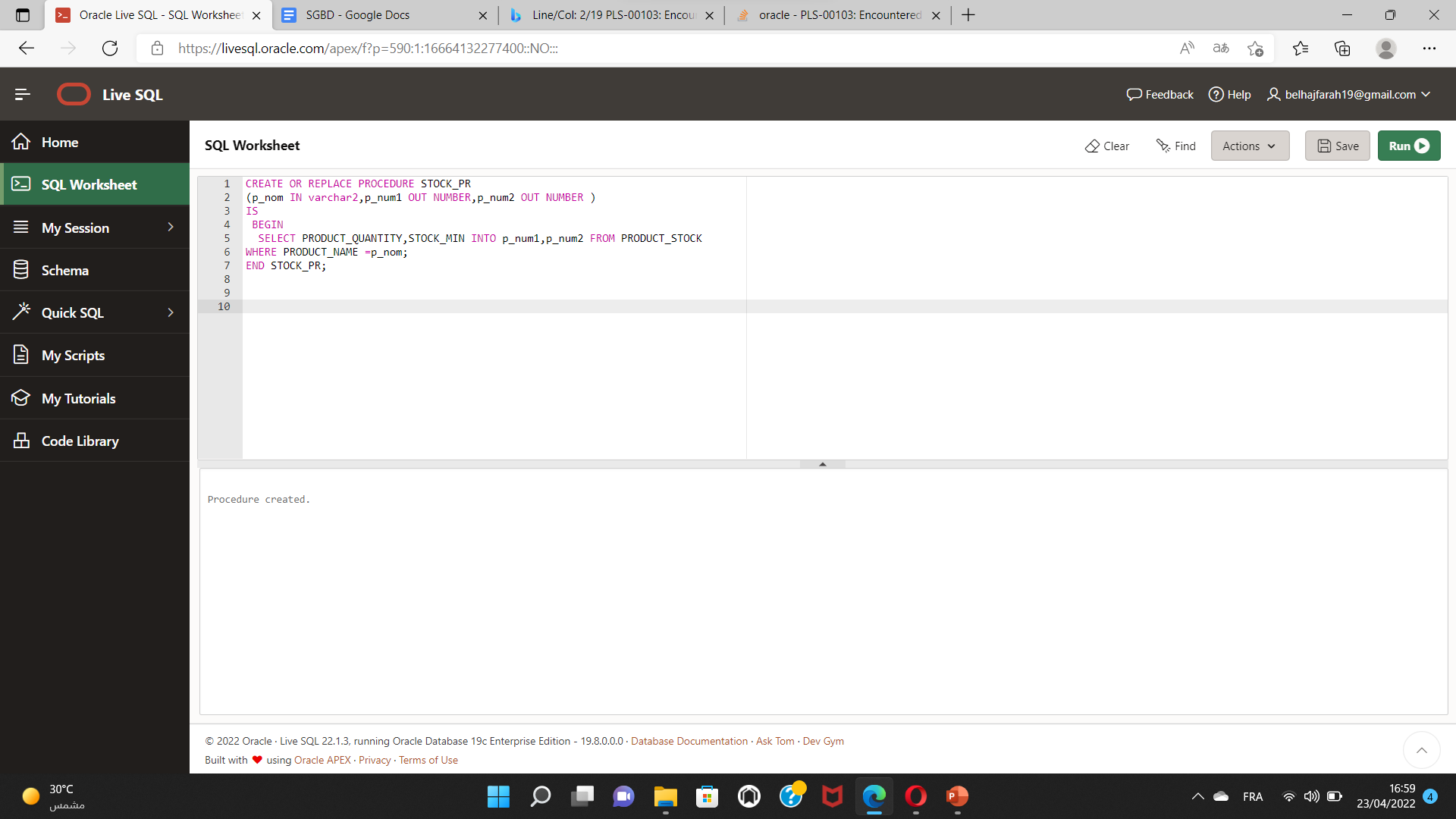
BEGIN

SELECT PRODUCT\_QUANTITY,STOCK\_MIN INTO p\_num1,p\_num2 FROM PRODUCT\_STOCK

WHERE PRODUCT\_NAME =p\_nom;

END STOCK\_PR;





**l’appel de la procédure:**

SET SERVEROUTPUT ON

ACCEPT S\_Nom PROMPT('saisir le nom de médicament');//saisie par user

DECLARE

P\_Nom varchar2(15):= &S\_Nom;

P\_Num1 NUMBER;

P\_Num2 NUMBER;

BEGIN

STOCK\_PR(P\_Nom,P\_Num1,P\_Num2);

DBMS\_OUTPUT.PUT\_LINE('le stock de produit recherché est '||P\_Num1||'et le stock min de ce produit est '||P\_Num2);

END;

/

**9-les fonctions:**

**Question:écrire une fonction qui retourne le nombre des clients dans une ville donnée**

create or replace function NB\_CLIENT

(p\_adr in VARCHAR2)

return NUMBER

is

v\_nb Number;

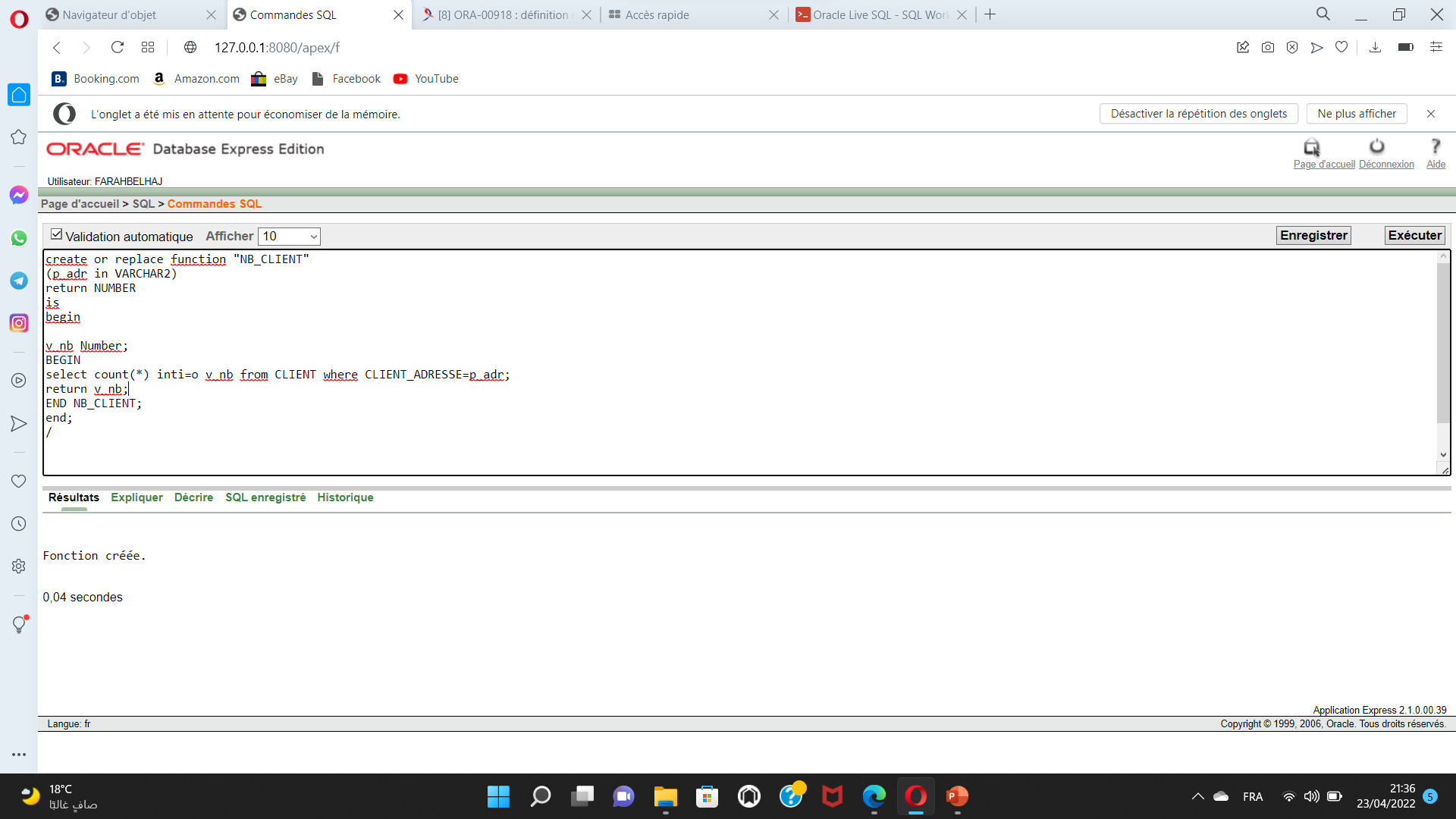
begin

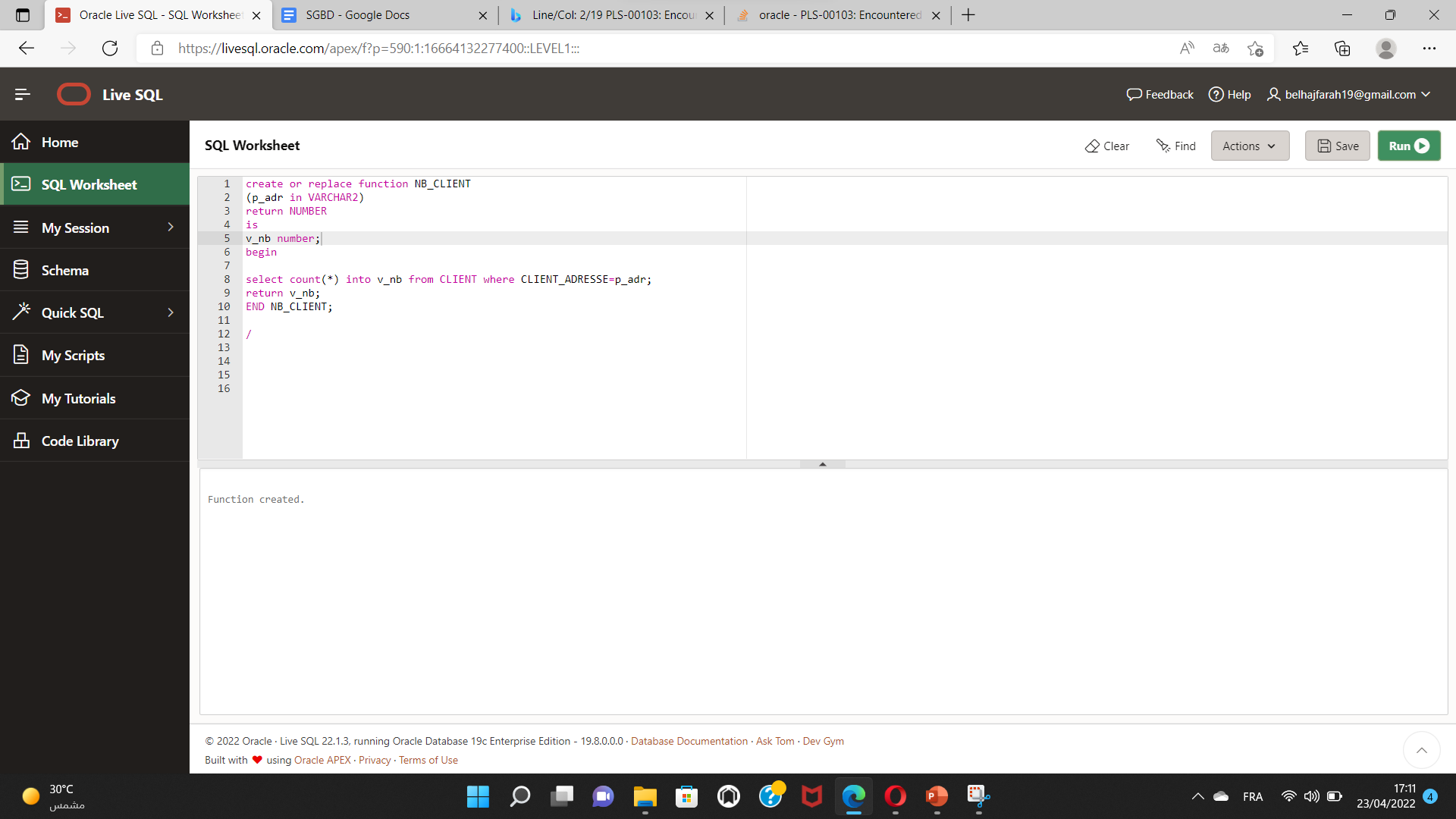
select count(\*) into v\_nb from CLIENT where CLIENT\_ADRESSE=p\_adr;

return v\_nb;

END NB\_CLIENT;

/

****

****

**l’appel de la fonction:**

set serveroutput on

Accept s\_adr Prompt (‘donner l’adresse du client à chercher);//saisie par user

DECLARE

V\_adr varchar2(15):=&s\_adr;

V\_nbr NUMBER;

BEGIN

V\_nbr:=NB\_CLIENT(V\_adr);

DBMS\_OUTPUT.PUT\_LINE('Le nombre des clients dans une ville donnée est '||V\_nbr);

END;

/

**l’appel de la fonction par initialisation:**

DECLARE

V\_adr varchar2(15):='tunis';

V\_nbr NUMBER;

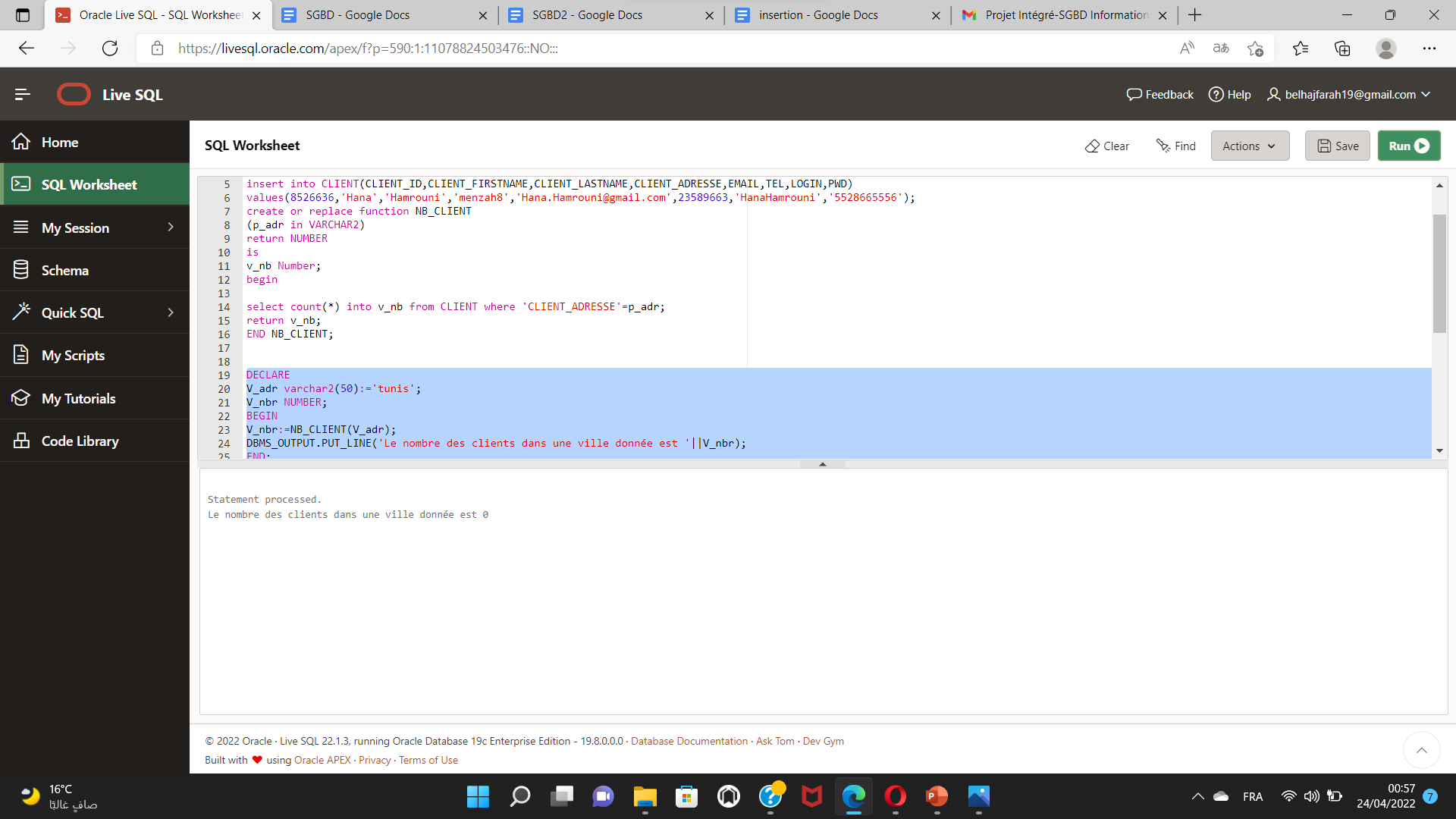
BEGIN

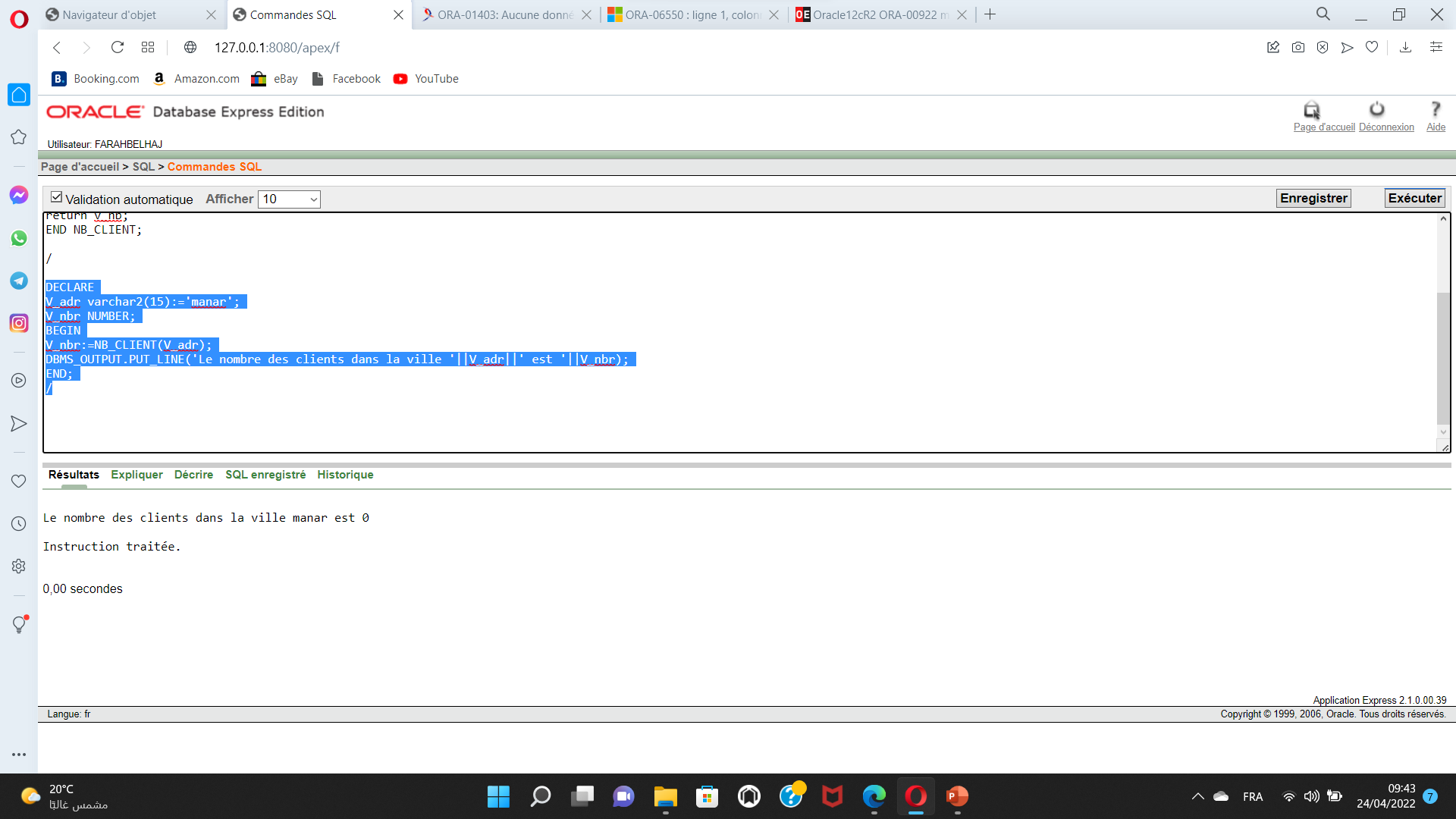
V\_nbr:=NB\_CLIENT(V\_adr);

DBMS\_OUTPUT.PUT\_LINE('Le nombre des clients dans la ville '||V\_adr||' est '||V\_nbr);

END;

/





**10-le trigger:**

**//on a créé un déclencheur sur la table ‘product\_stock’**

CREATE OR REPLACE TRIGGER trig\_stock

AFTER INSERT OR DELETE OR UPDATE ON PRODUCT\_STOCK

FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO trig\_stock

VALUES(‘ins’,SYSDATE,null,:NEW.PRODUCT\_QUANTITY);

ELSIF UPDATING THEN

INSERT INTO trig\_stock

VALUES(‘mod’,SYSDATE,:OLD.PRODUCT\_QUANTITY,:NEW.PRODUCT\_QUANTITY);

ELSE

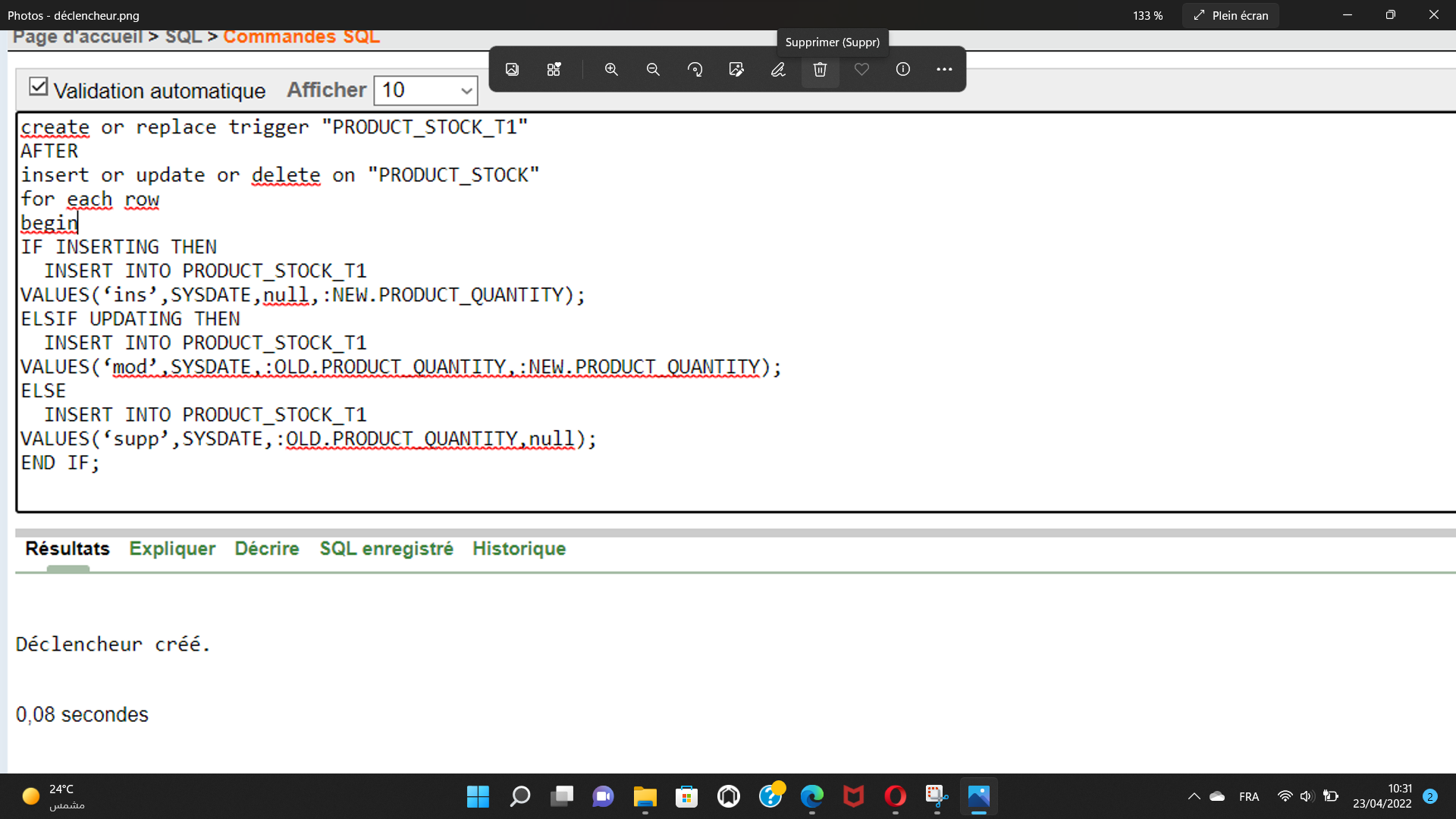
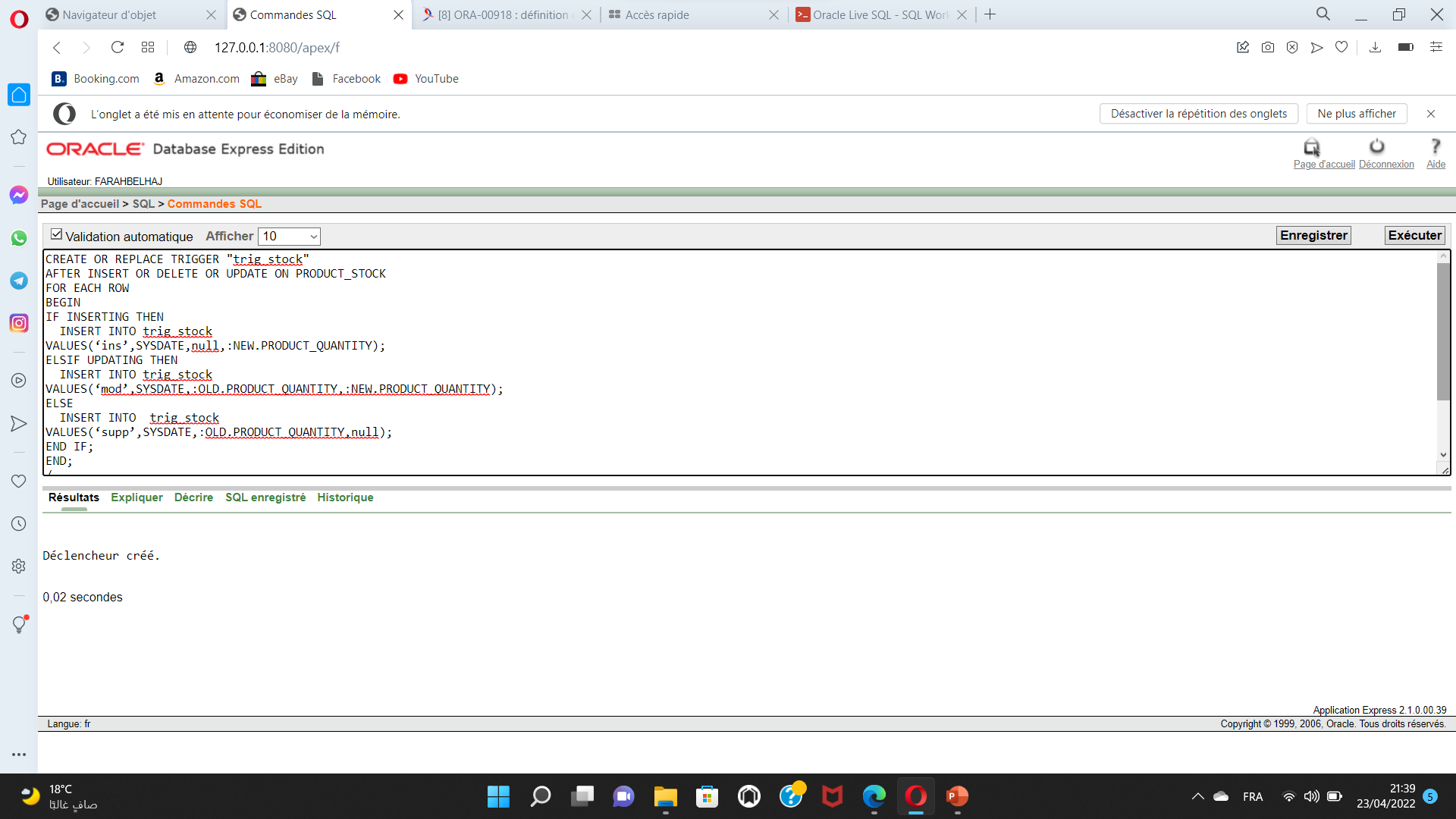
INSERT INTO trig\_stock

VALUES(‘supp’,SYSDATE,:OLD.PRODUCT\_QUANTITY,null);

END IF;

END;

/



**11-les curseurs implicites**

**a-calculer la somme des salaires des responsable\_STOCK**

DECLARE

V\_sum NUMBER;

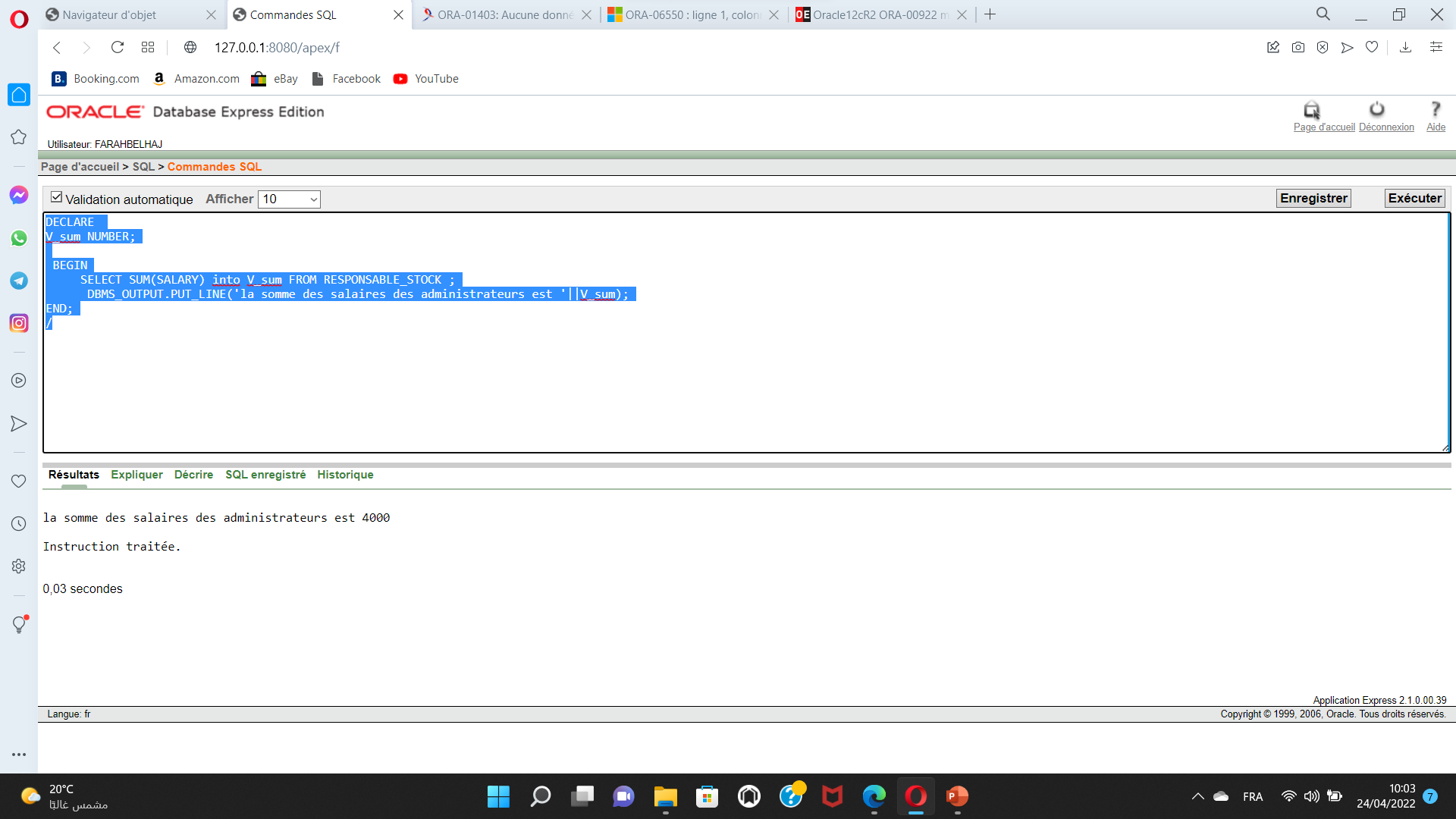
BEGIN

SELECT SUM(SALARY) into V\_sum FROM RESPONSABLE\_STOCK ;

DBMS\_OUTPUT.PUT\_LINE('la somme des salaires des administrateurs est '||V\_sum);

END;

/

****

**b-calculer la somme des salaires des responsable\_STOCK et des administrateurs**

DECLARE

V\_sum1 NUMBER;

V\_sum2 NUMBER;

BEGIN

SELECT SUM(SALARY) into V\_sum1 FROM RESPONSABLE\_STOCK

UNION

SELECT SUM(SALARY) into V\_sum2 FROM ADMINISTRATEUR;

DBMS\_OUTPUT.PUT\_LINE('la somme des salaires des administrateurs est '||V\_sum1+V\_sum2);

END;

/

**12-les curseurs explicites**

**Question: afficher les clients dont le mode de paiement est cash**

DECLARE

cursor user\_cursor is select CLIENT.CLIENT\_FIRSTNAME CN,CLIENT.CLIENT\_LASTNAME CP from CLIENT,COMMANDE where PAIMENT\_MODE='cash' and COMMANDE.CLIENT\_ID=CLIENT.CLIENT\_ID;

user\_record user\_cursor% ROWTYPE;

BEGIN

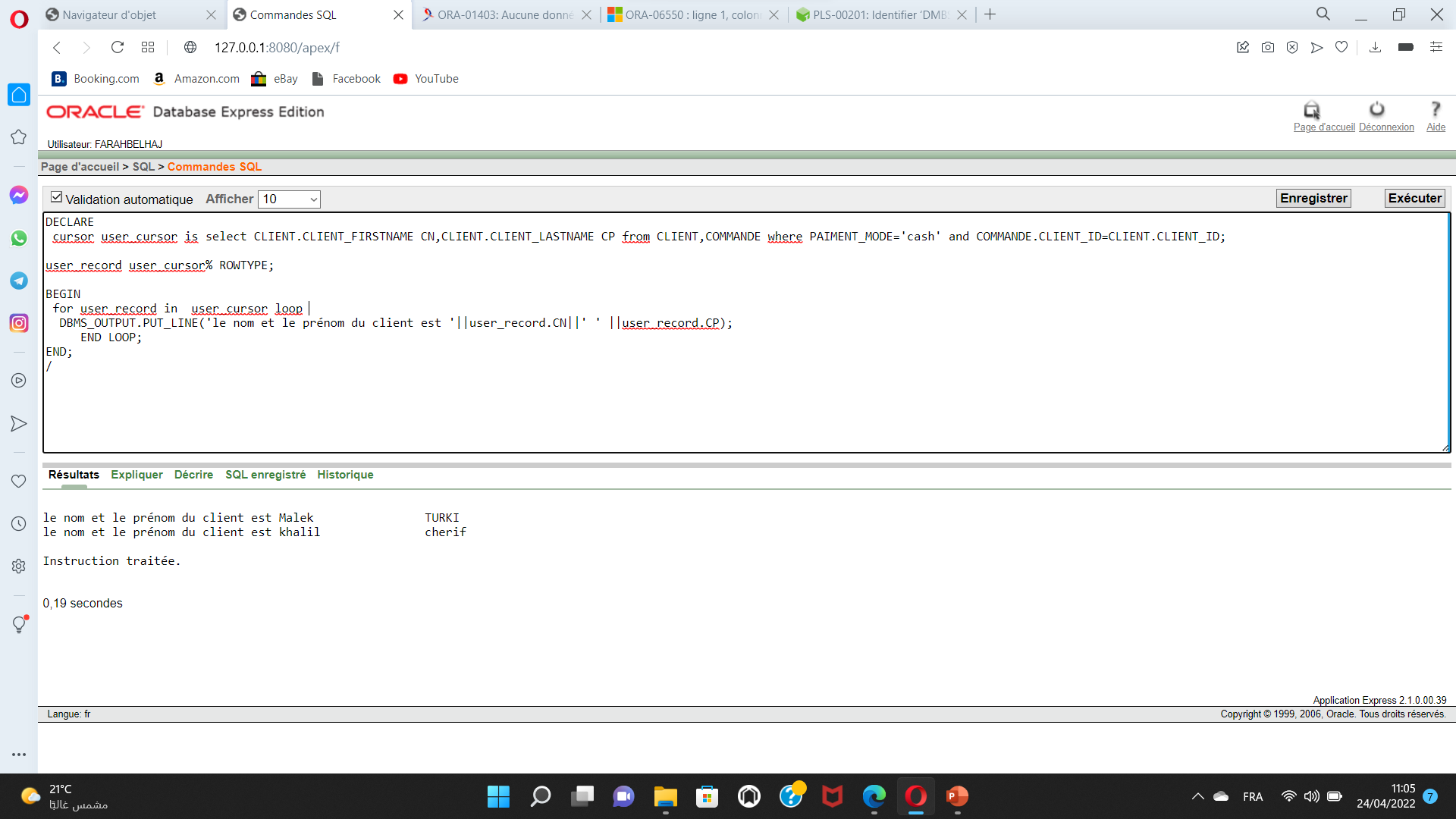
for user\_record in user\_cursor loop

DBMS\_OUTPUT.PUT\_LINE('le nom et le prénom du client est '||user\_record.CN||' ' ||user\_record.CP);

END LOOP;

END;

/

****

**13-les curseurs explicites paramétré**

**Question: Afficher les clients dans l’adresse manar suivi par leur nombre ,et faire le même travail pour les clients d’adresse menzah8**

DECLARE

cursor client\_cursor (p\_CLIENT\_ADRESSE in ) IS SELECT CLIENT.CLIENT\_ID,CLIENT.CLIENT\_FIRSTNAME,CLIENT.CLIENT\_LASTNAME FROM CLIENT WHERE CLIENT.CLIENT\_ADRESSE=p\_CLIENT\_ADRESSE;

client\_record client\_cursor%ROWTYPE;

BEGIN

OPEN client\_cursor('manar');

loop

fetch client\_cursor into client\_record;

exit when client\_cursor % NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('le nom et le prénom de client de manar est :'||client\_record.CLIENT\_FIRSTNAME ||''||client\_record.CLIENT\_LASTNAME);

END loop;

DBMS\_OUTPUT.PUT\_LINE('le nombre des clients est '||client\_record%ROWCOUNT);

CLOSE client\_cursor;

OPEN client\_cursor(menzah8);

loop

fetch client\_cursor into client\_record;

exit when client\_cursor %NOT FOUND;

DBMS\_OUTPUT.PUT\_LINE('le nom et le prénom de client de manar est :'||client\_record.CLIENT\_FIRSTNAME ||''||client\_record.CLIENT\_LASTNAME);

END loop;

DBMS\_OUTPUT.PUT\_LINE('le nombre des clients est '||client\_record%ROWCOUNT);

CLOSE client\_cursor;