Rapport TP ASGBD

Nom : Zait

Prénom :Fouad

Section: SII

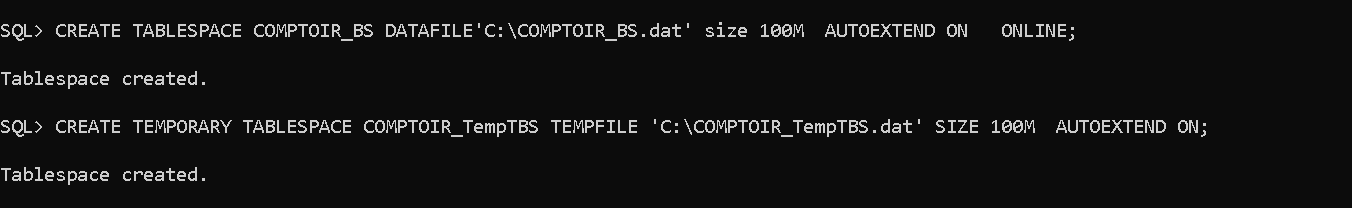
Matricule:181831072145

TP1:

1- Créer deux Tablespaces**COMPTOIRS\_BS** *et* **COMPTOIRS\_TempTBS :**

**CREATE TABLESPACE COMPTOIR\_BS DATAFILE'C:\COMPTOIR\_BS.dat' size 100M AUTOEXTEND ON ONLINE;**

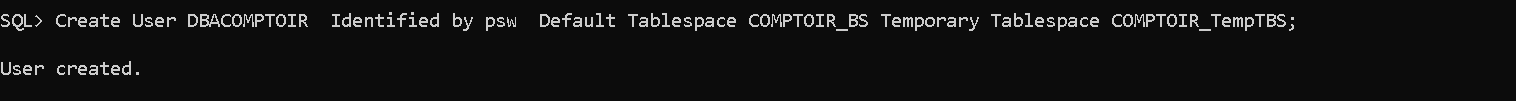
**CREATE TEMPORARY TABLESPACE COMPTOIR\_TempTBS TEMPFILE 'C:\COMPTOIR\_TempTBS.dat' SIZE 100M AUTOEXTEND ON;**

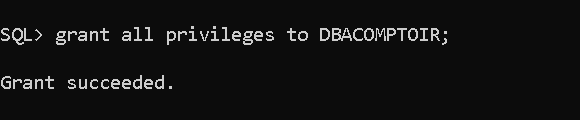


2- Créer un utilisateur **DBACOMPTOIRS**  en lui attribuant les deux tablespaces créés précédemment et 3-Donner tous les privilèges à cet utilisateur.

**Create User DBACOMPTOIR Identified by psw Default Tablespace COMPTOIR\_BS Temporary Tablespace COMPTOIR\_TempTBS;**

**grant all privileges to DBACOMPTOIR;**





4- Créer les relations de la base de données avec toutes les contraintes d’intégrité :

CREATE TABLE DETAILCOMMANDE(

NOCOM NUMBER(38),

REFPROD NUMBER(38),

PRIXUNIT NUMBER(38),

QTE NUMBER(38),

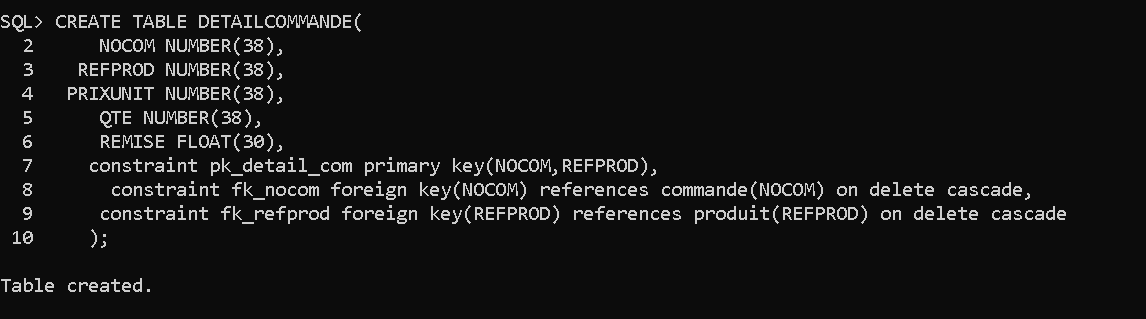
REMISE FLOAT(30),

constraint pk\_detail\_com primary key(NOCOM,REFPROD),

constraint fk\_nocom foreign key(NOCOM) references commande(NOCOM) on delete cascade,

constraint fk\_refprod foreign key(REFPROD) references produit(REFPROD) on delete cascade

);



CREATE TABLE categorie(

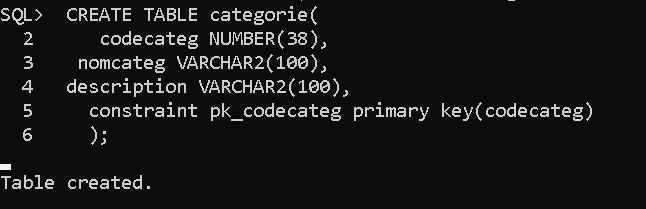
codecateg NUMBER(38),

nomcateg VARCHAR2(100),

description VARCHAR2(100),

constraint pk\_codecateg primary key(codecateg)

);



CREATE TABLE client(

codecli VARCHAR2(100),

societe VARCHAR2(100),

contact VARCHAR2(100),

fonction VARCHAR2(100),

adresse VARCHAR2(100),

ville VARCHAR2(100),

region VARCHAR2(100),

codepostal VARCHAR2(100),

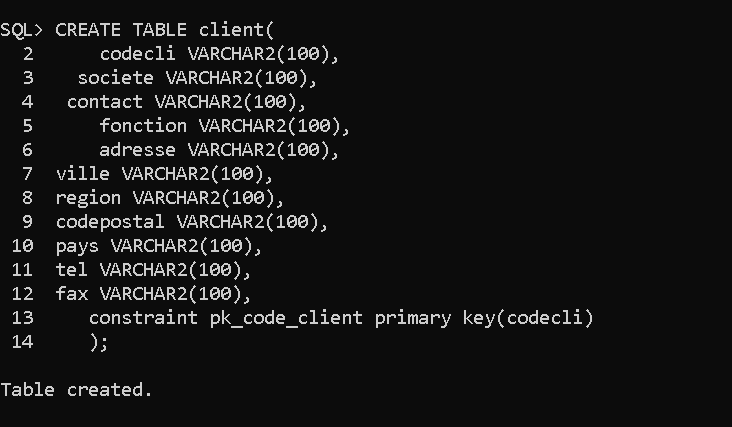
pays VARCHAR2(100),

tel VARCHAR2(100),

fax VARCHAR2(100),

constraint pk\_code\_client primary key(codecli)

);



CREATE TABLE fournisseur(

nofour NUMBER(38),

societe VARCHAR2(100),

contact VARCHAR2(100),

fonction VARCHAR2(100),

adresse VARCHAR2(100),

ville VARCHAR2(100),

region VARCHAR2(100),

codepostal VARCHAR2(100),

pays VARCHAR2(100),

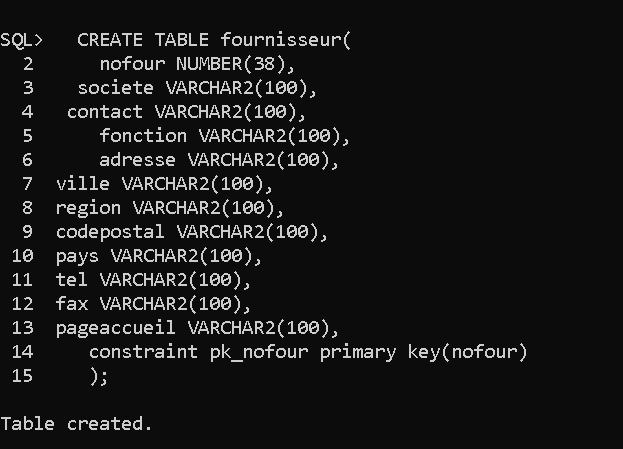
tel VARCHAR2(100),

fax VARCHAR2(100),

pageaccueil VARCHAR2(100),

constraint pk\_nofour primary key(nofour)

);



CREATE TABLE produit(

refprod NUMBER(38),

nomprod VARCHAR2(100),

nofour NUMBER(38),

codecateg NUMBER(38),

qteparunit VARCHAR2(100),

prixunit NUMBER(38),

unitesstock NUMBER(38),

unitescom NUMBER(38),

niveaureap NUMBER(38),

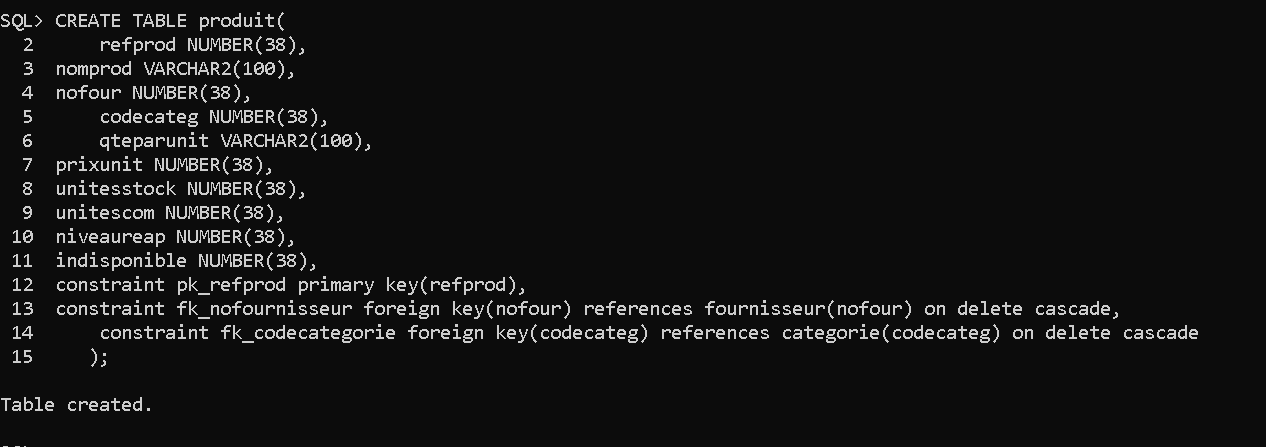
indisponible NUMBER(38),

constraint pk\_refprod primary key(refprod),

constraint fk\_nofournisseur foreign key(nofour) references fournisseur(nofour) on delete cascade,

constraint fk\_codecategorie foreign key(codecateg) references categorie(codecateg) on delete cascade

);



CREATE TABLE messager(

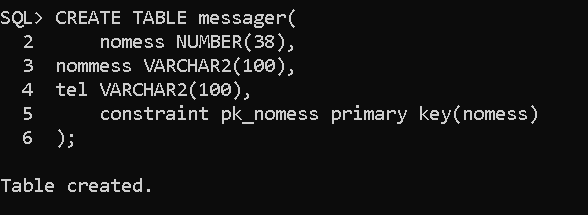
nomess NUMBER(38),

nommess VARCHAR2(100),

tel VARCHAR2(100),

constraint pk\_nomess primary key(nomess)

);



CREATE TABLE employe(

noemp NUMBER(38),

nom VARCHAR2(100),

prenom VARCHAR2(100),

fonction VARCHAR2(100),

titrecourtoisie VARCHAR2(100),

datenaissance VARCHAR2(100),

dateembauche VARCHAR2(100),

adresse VARCHAR2(100),

ville VARCHAR2(100),

region VARCHAR2(100),

codepostal VARCHAR2(100),

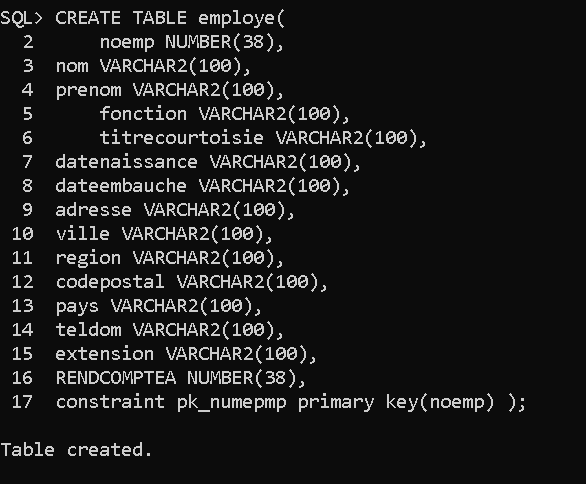
pays VARCHAR2(100),

teldom VARCHAR2(100),

extension VARCHAR2(100),

RENDCOMPTEA NUMBER(38),

constraint pk\_numepmp primary key(noemp) );



CREATE TABLE commande(

nocom NUMBER(38),

codecli VARCHAR2(100),

noemp NUMBER(38),

datecom DATE,

alivavant DATE,

dateenv DATE,

nomess NUMBER(38),

port VARCHAR2(100),

destinataire VARCHAR2(100),

adrliv VARCHAR2(100),

villeliv VARCHAR2(100),

regionliv VARCHAR2(100),

codepostalliv VARCHAR2(100),

paysliv VARCHAR2(100),

constraint pk\_numcom primary key(nocom),

constraint fk\_nummessager foreign key(nomess) references messager(nomess) on delete cascade,

constraint fk\_numemp foreign key(noemp) references employe(noemp) on delete cascade,

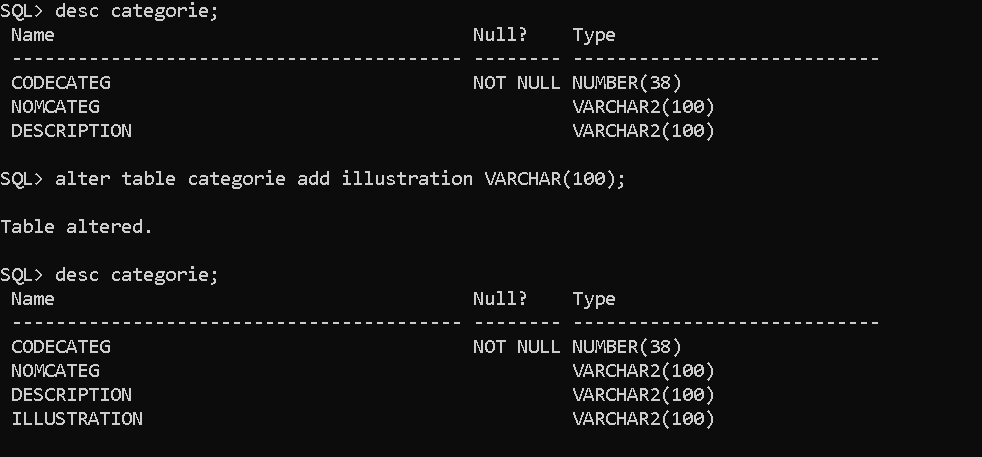
constraint fk\_codecli foreign key(codecli) references client(codecli) on delete cascade

);



5- Ajouter l’attribut ILLUSTRATION de type chaîne de caractères dans la relation CATEGORIE :

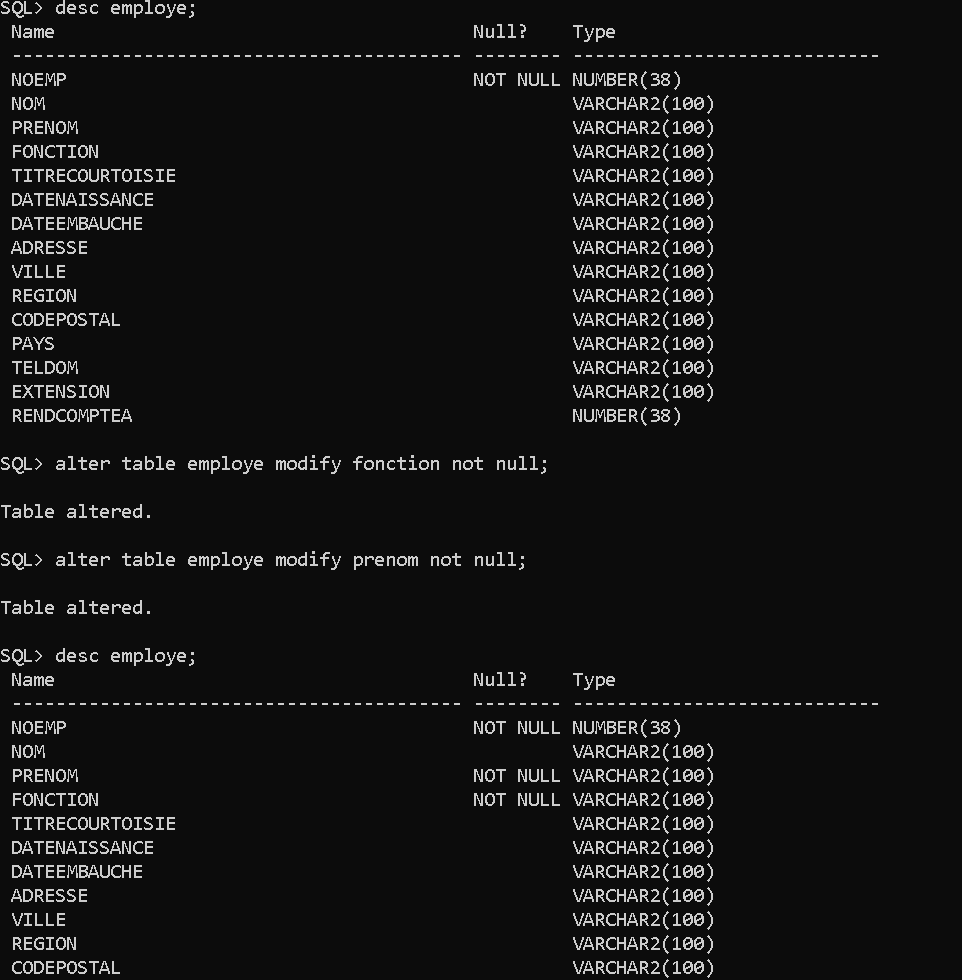
**alter table categorie add illustration VARCHAR(100);**



6- Ajouter la contrainte not null pour les attributs FONCTION et PRENOM de la relation EMPLOYE.

**alter table employe modify fonction not null;**

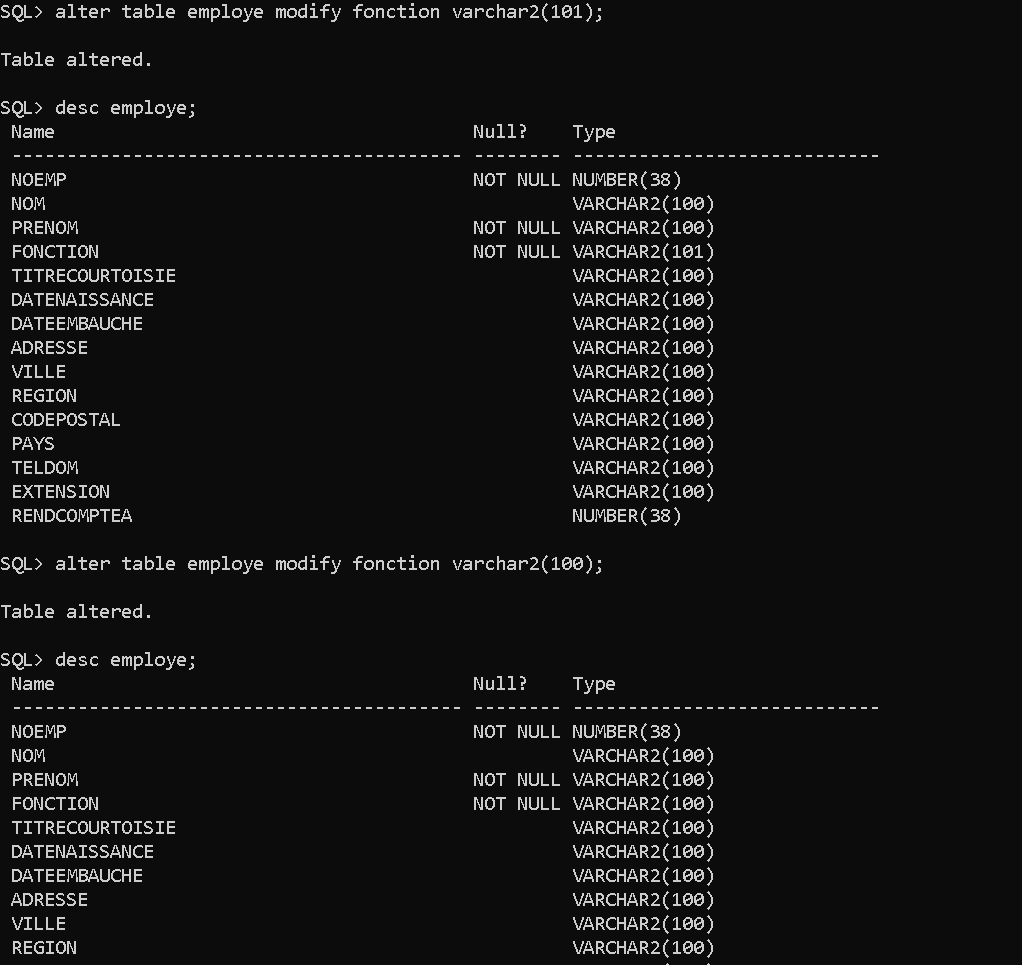
**alter table employe modify prenom not null;**



7- Modifier la longueur de l’attribut FONCTION (agrandir, réduire).

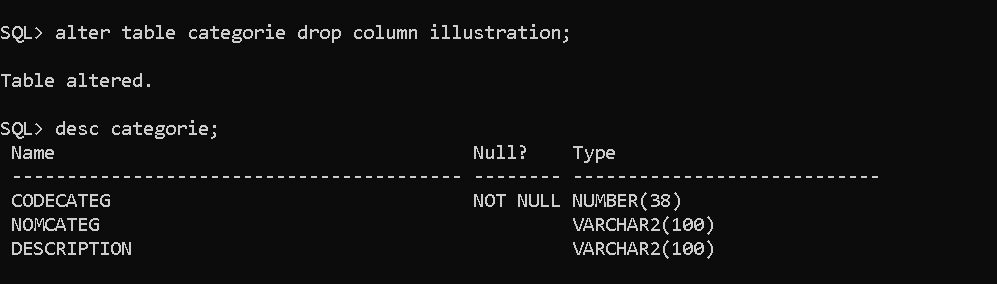
**alter table employe modify fonction varchar2(101);**

**alter table employe modify fonction varchar2(100);**



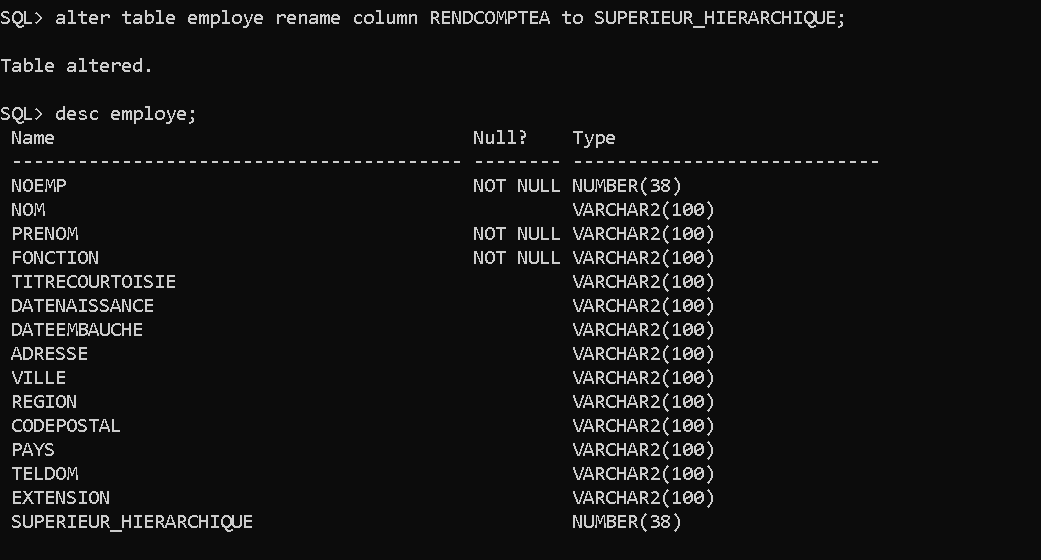
8- Supprimer la colonne ILLUSTRATION dans la table CATEGORIE. Vérifier la suppression.

**alter table categorie drop column illustration;**



9- Renommer la colonne RENDCOMPTEA dans la table EMPLOYE par SUPERIEUR\_HIERARCHIQUE. Vérifier.

**alter table employe rename column RENDCOMPTEA to SUPERIEUR\_HIERARCHIQUE;**



10- Ajouter la contrainte suivante sur la table commande : la date de la commande < date d’envoie < date à livrer avant.

**alter table commande add constraint cons check(DATECOM < DATEENV AND DATEENV <ALIVAVANT);**

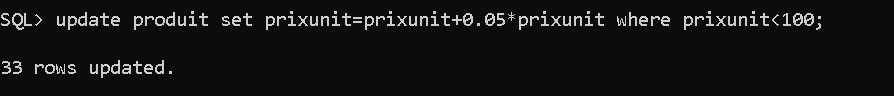


11- Remplir toutes les tables par les instances en exécutant le script insert.sql. Quels sont les problèmes rencontrés.

**@insert.sql**

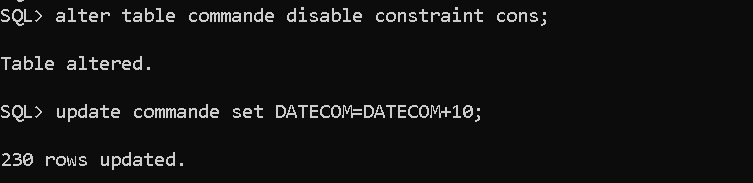
12- Supposons que l'on veut augmenter de 5% le prix des produits dont le prix actuel est inférieur à 100. Que faut-il faire ?

update produit set prixunit = prixunit+( prixunit\*0.05) where prixunit< 100;

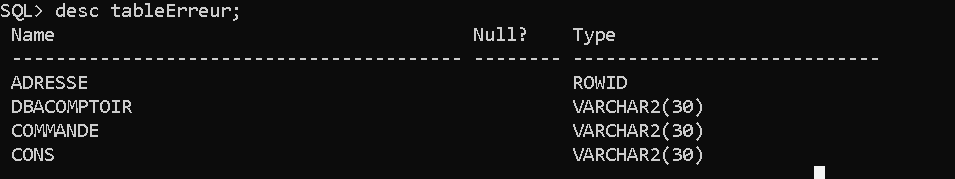


13- Augmenter la date de la commande par 10 jours. Désactiver la contrainte pour autoriser la modification. Réactiver la contrainte.

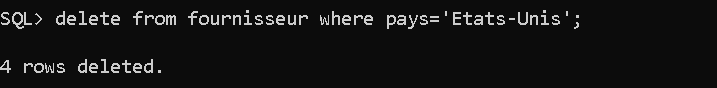
Désactivation de la contrainte:



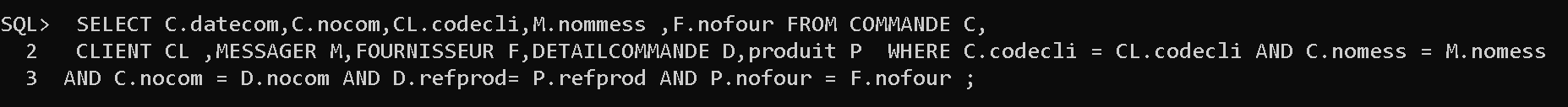
Réactivation de la contrainte:



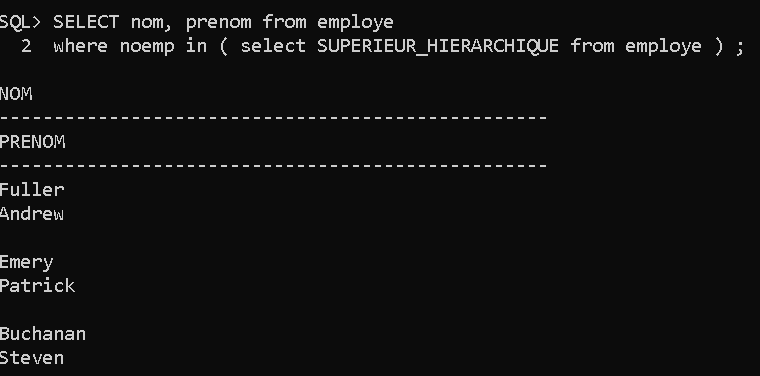
14- Supprimer tous les fournisseurs de pays «Etats-Unis ». Quels sont les problèmes rencontrés.



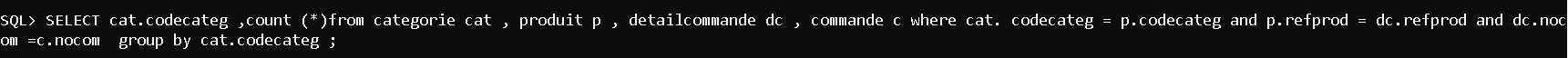
15- On veut connaître pour chaque commande, la date de commande, le n° de commande, le nom du client, le nom du messager, le nom du fournisseur.



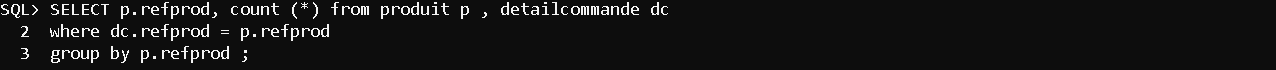
16- On veut connaître pour chaque employé le nom et le prénom de son supérieur hiérarchique.



17- Pour chaque catégorie, donner le nombre de commandes.



18- Calculer le chiffre d'affaires pour chaque produit commandé par les clients.



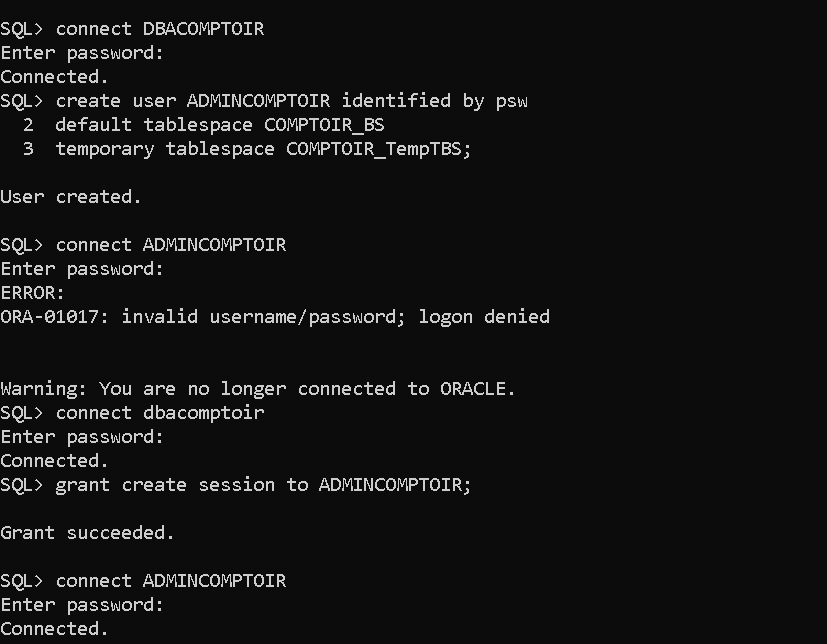
###### TP N°2 : **Privilèges d'accès à la base de données**

1- Connectez-vous avec l’utilisateur **DBACOMPTOIRS.**

2- Créez un autre utilisateur : **ADMINCOMPTOIRS** qui possède les mêmes tablespaces.

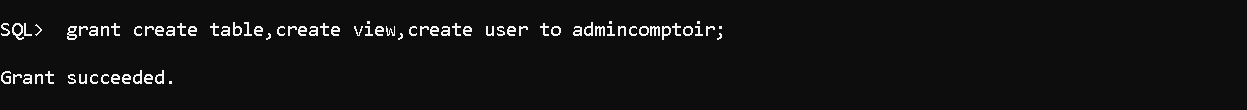
3-Connectez-vous à l’aide de cet utilisateur. Que remarquez-vous ?

4-Donnez le droit de création d’une session pour cet utilisateur (Create Session) et reconnectez-vous.



Pour la question 3 On remarque que nous ne pouvons pas connecter à cet utilisateur car il manque le privilège (create session)

5- Donnez les privilèges suivants à **ADMINCOMPTOIRS**: créer des tables, des vues et des utilisateurs. Vérifiez.

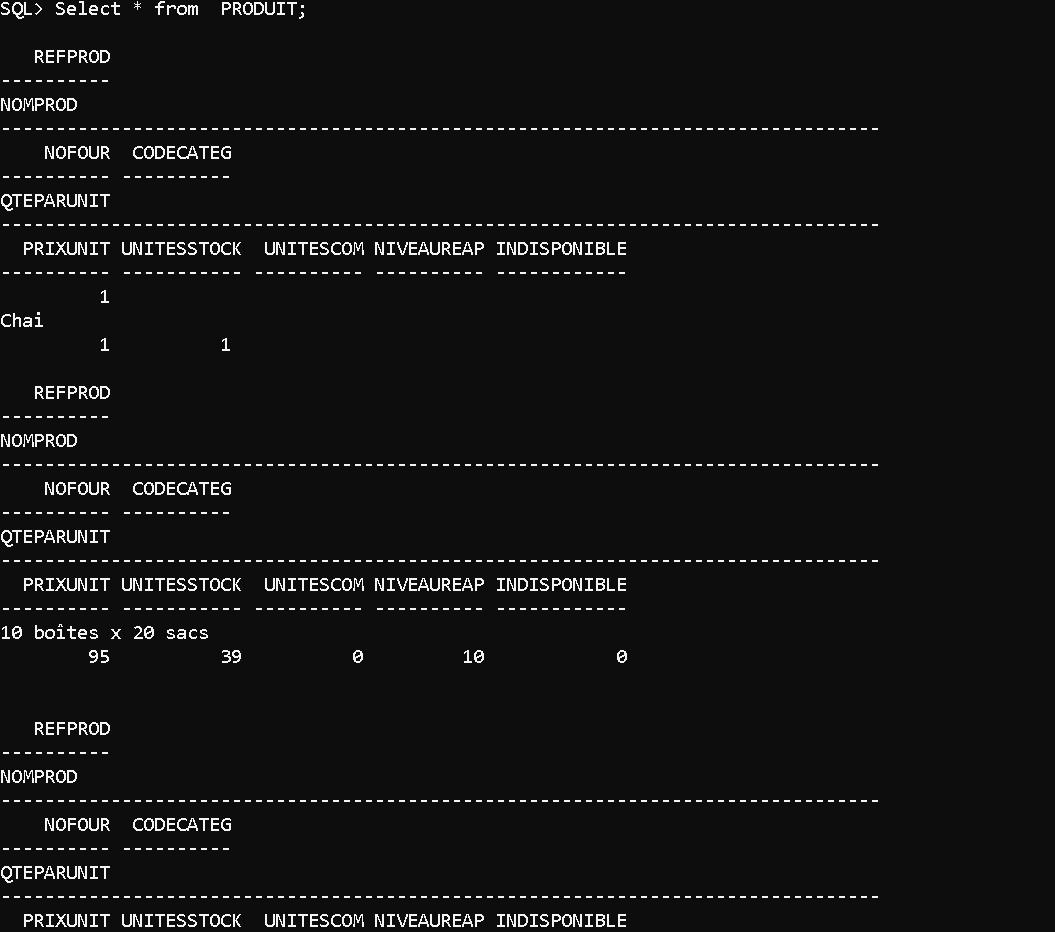
6- Exécutez la requête Q1 suivante : **Select \* from PRODUIT;** Que remarquez-vous ?

À l’execution de select on remarque une erreur car on n’a pas donner le droit de lecture a admincomptoir.

7- Donnez le droit de lecture à cet utilisateur pour la table **PRODUIT**. Exécutez la requête Q1 maintenant.



Verification:



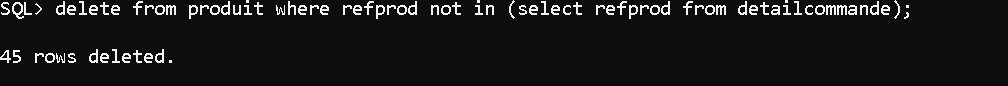
8- On veut supprimer toutes les produits qui ne sont plus vendus. Que faut-il faire ? Que remarquez-vous ?

Un message d’erreur sera afficher , On doit donner le droit de lecture (SELECT) sur la table DETAILCOMMANDE a l’utilisateur ADMINCOMPTOIRS et on doit lui donner aussi le droit de suppression pour la table produit .

9- Donner le droit de suppression à cet utilisateur pour la table **PRODUIT** et réessayez de refaire la suppression.



Réessayer la suppression:



10- Créez un index NP\_IX sur l’attribut NOMPROD de la table **PRODUIT**. Que remarquez-vous ?

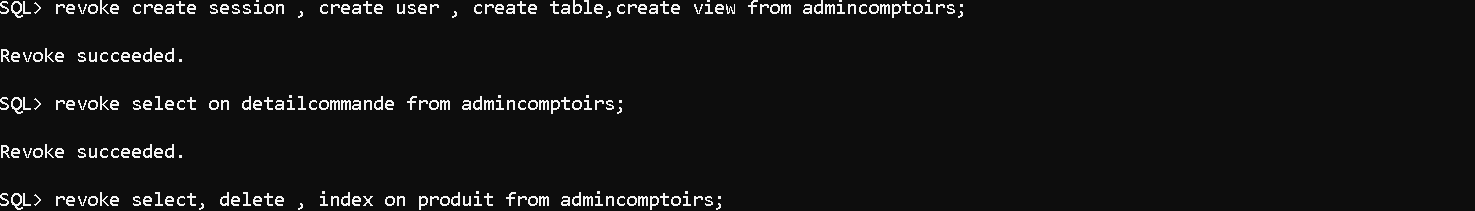
Quand on execute create index NP\_IX on produit(nomprod); un message d’erreur est affiché car on a pas le privilege de créer un index.

11- Donnez le droit de création d’index à **ADMINCOMPTOIRS** pour la table **PRODUIT**, ensuite réessayez de créer l’index. Que se passe-t-il ?





12-Enlevez les privilèges précédemment accordés.



13- Vérifiez que les privilèges ont bien été supprimés.



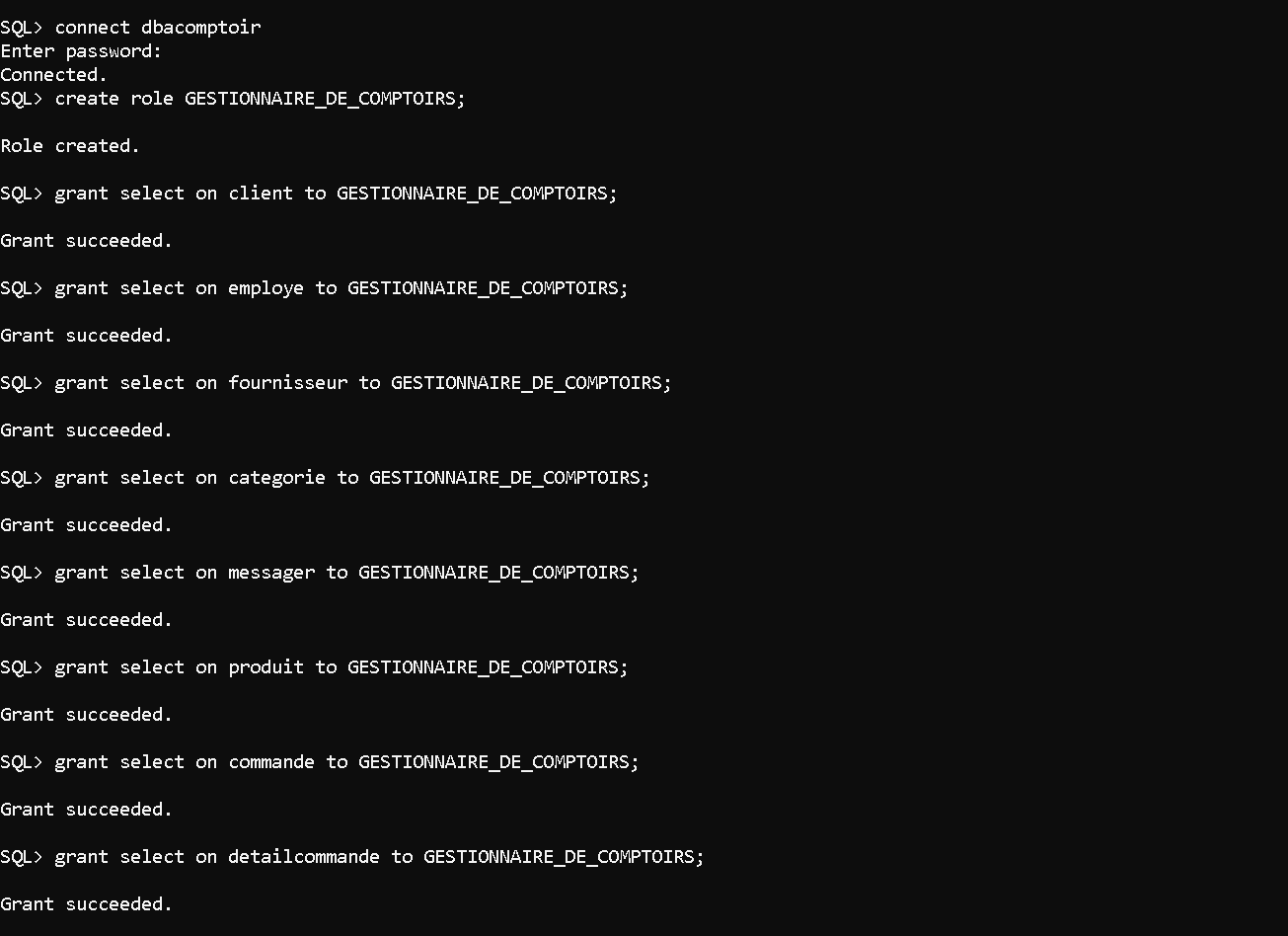
14- Créez un profil « **Comptoirs\_Profil** » qui est caractérisé par : ( 3 sessions simultanés autorisées, Un appel système ne peut pas consommer plus de 20 secondes de CPU, Chaque session ne peut excéder 60 minutes, Un appel système ne peut lire plus de 1200 blocs de données en mémoire et sur le disque, Chaque session ne peut allouer plus de 40 ko de mémoire en SGA, Pour chaque session, 15 minutes d’inactivité maximum sont autorisés, 5 tentatives de connexion avant blocage du compte, Le mot de passe est valable pendant 70 jours et il faudra attendre 50 jours avant qu’il puisse être utilisé à nouveau, 1 seul jour d’interdiction d’accès après que les 5 tentatives de connexion ont été atteintes, La période de grâce qui prolonge l’utilisation du mot de passe avant son changement est de 5 jours).



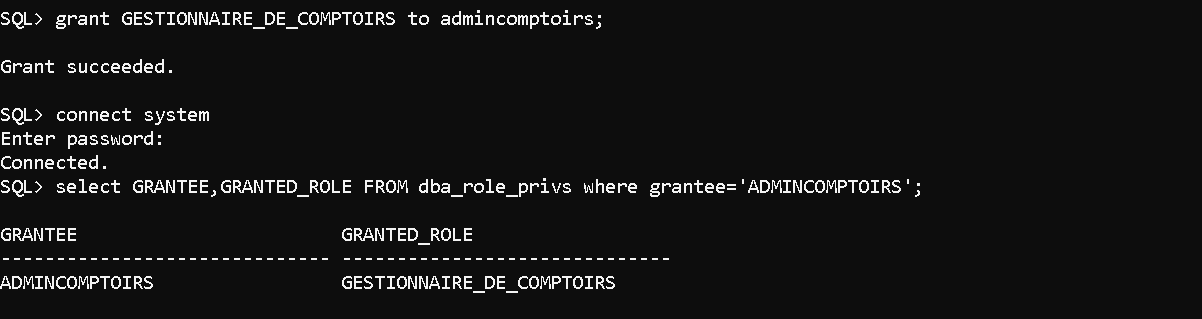
15- Affectez ce profil à l’utilisateur **ADMINCOMPTOIRS**. Vérifiez.



16- Créez le rôle : « **GESTIONNAIRE\_DE\_COMPTOIRS**» qui peut voir les tables **FOURNISSEUR, CLIENT, EMPLOYE, CATEGORIE, MESSAGER** et peut modifier les tables **PRODUIT, COMMANDE ET DETAILCOMMANDE.**

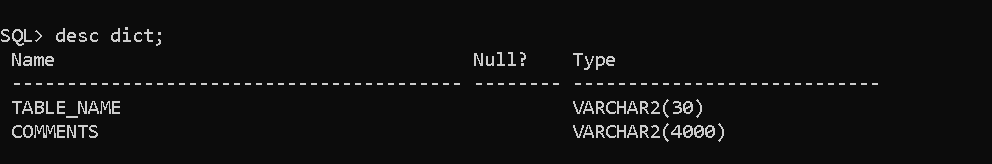
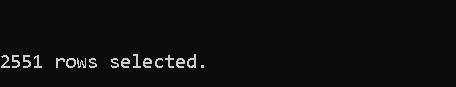
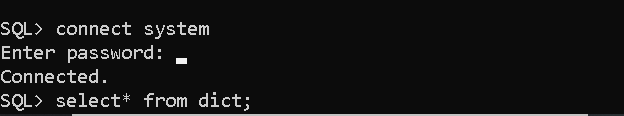


17- Assignez ce rôle à **ADMINCOMPTOIRS**. Vérifier que les autorisations assignées au rôle **GESTIONNAIRE\_ DE\_COMPTOIRS**, ont été bien transférées sur l’utilisateur à **ADMINCOMPTOIRS**.



TP N°3: **Dictionnaire Oracle**

1- Connecter en tant que « **System** ». Lister le catalogue « DICT ». Il contient combien d’instances ? Donner sa structure ?



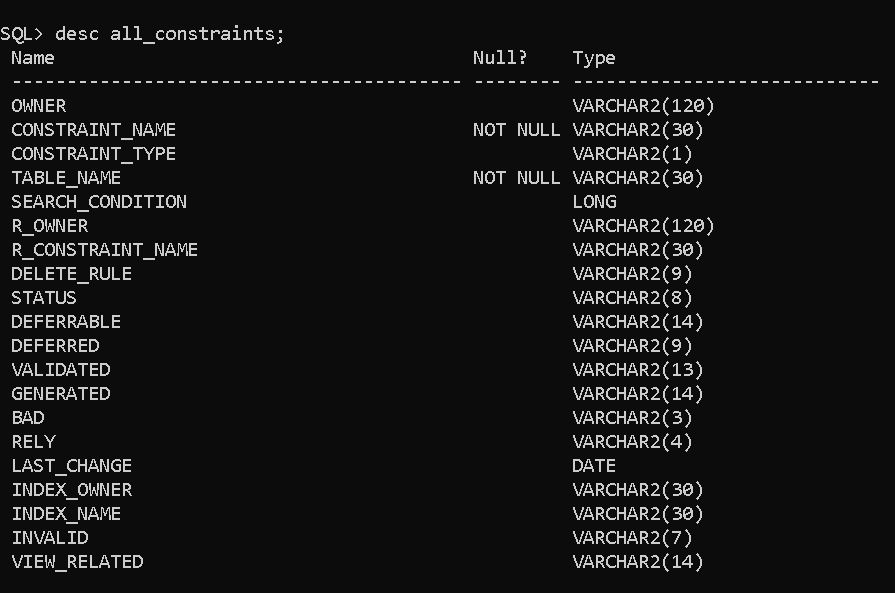
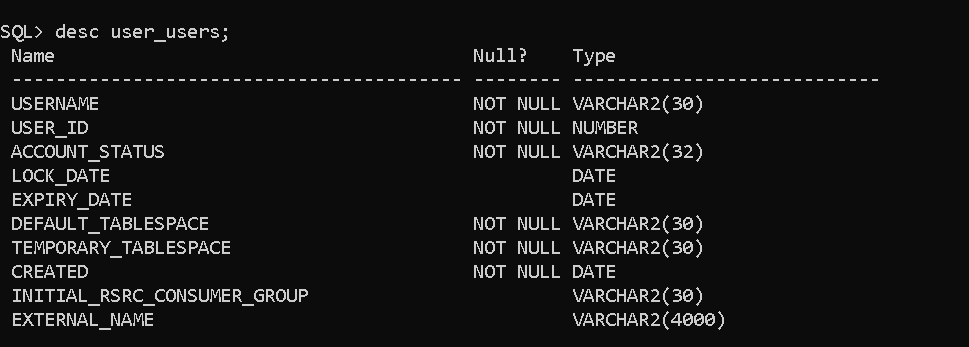
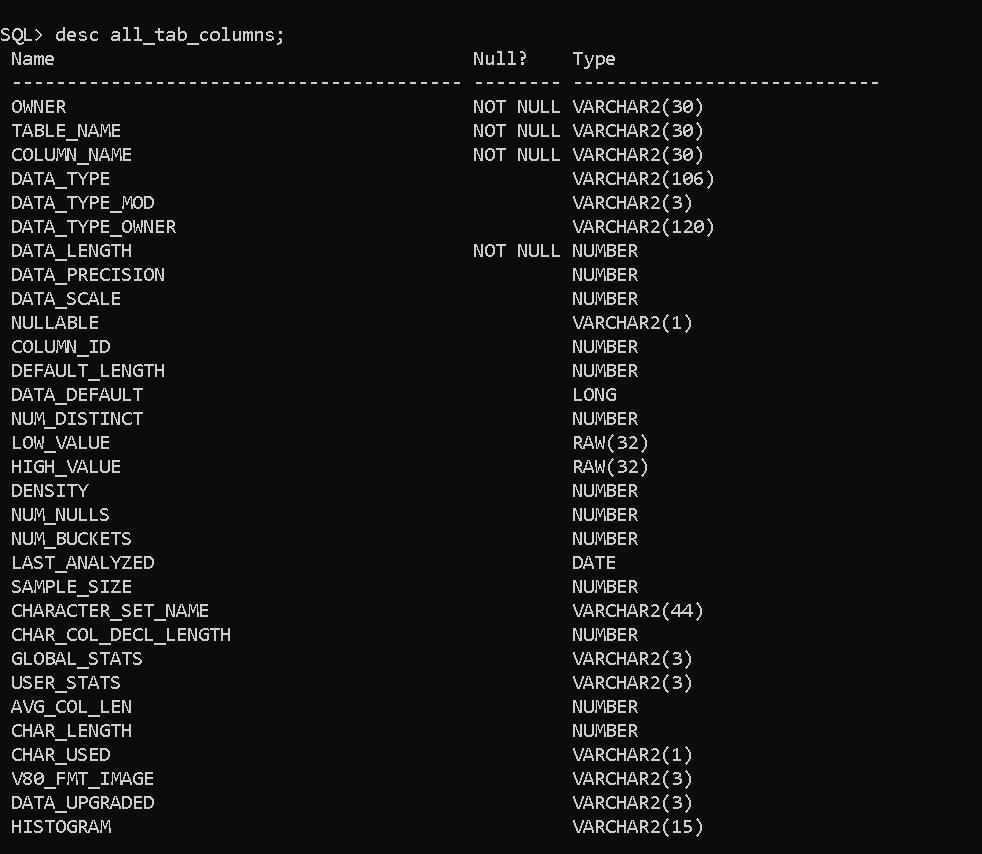
2- Donner le rôle et la structure des tables (ou vues) suivantes : ALL\_TAB\_COLUMNS, USER\_USERS, ALL\_CONSTRAINTS et USER\_TAB\_PRIVS.

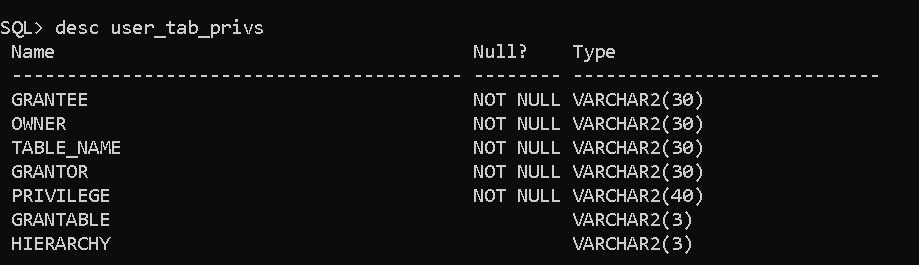
**ALL\_TAB\_COLUMNS** :regroupe les colonnes des tables de la Base de données.

**USER\_USERS** : Contient des informations sur l’utilisateur courant.

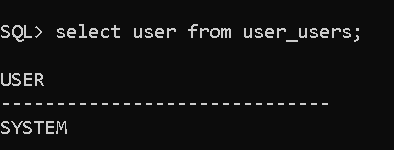
**ALL\_CONSTRAINTS** : Contient des informations sur toutes les contraintes de la Base de données.

**USER\_TAB\_PRIVS** : Regroupe les informations sur les privilèges des utilisateurs sur les tables de la base de données.



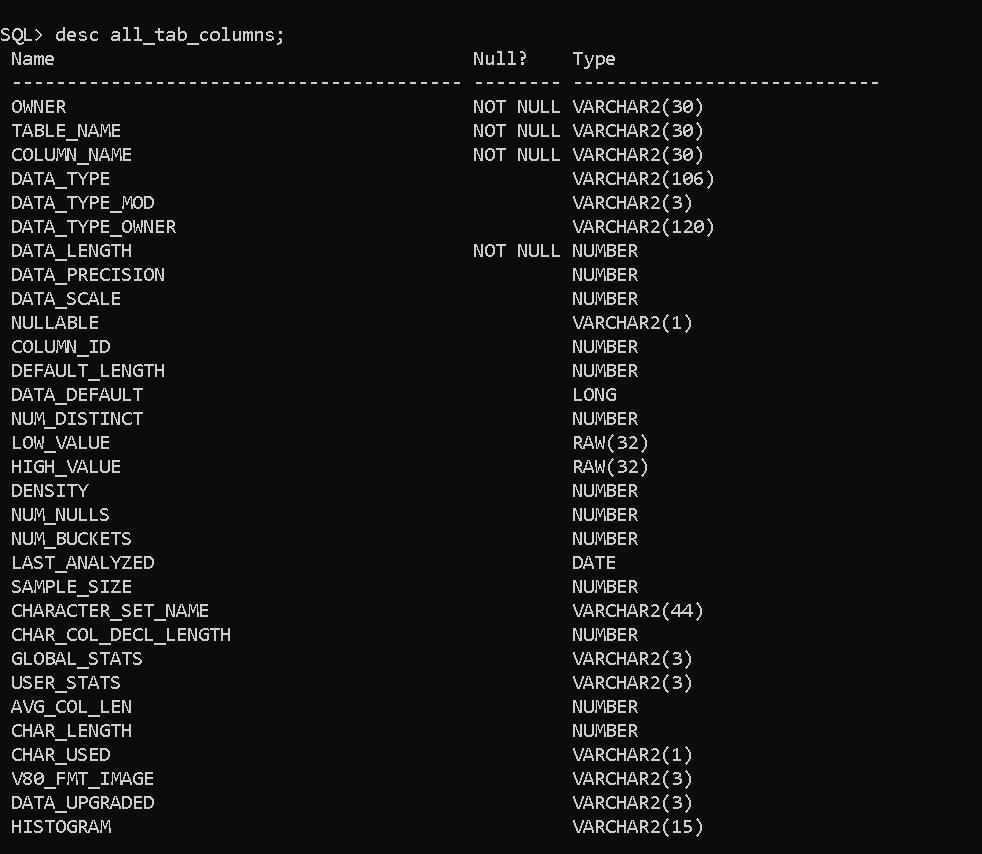


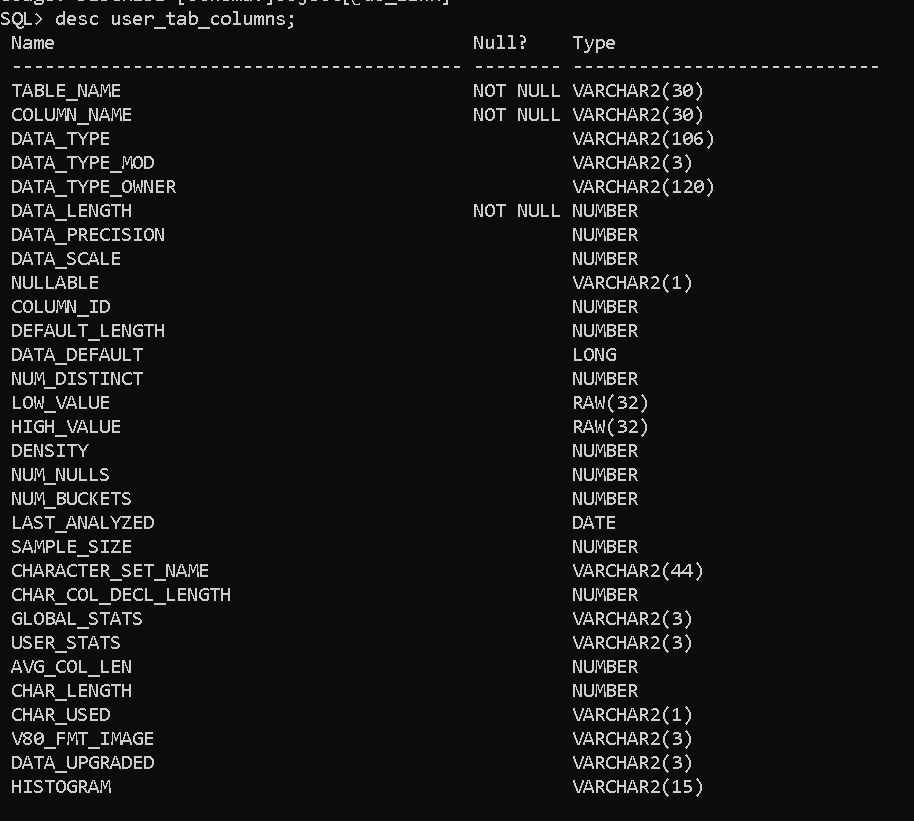
3-Trouver le nom d’utilisateur avec lequel vous êtes connecté ?



4- Comparer la structure et le contenu des tables ALL\_TAB\_COLUMNS et USER\_ TAB\_COLUMNS ?

les deux tables contiennent les mêmes informations sauf que ALL\_TAB\_COLUMNS contient OWNER en plus.





5- Vérifiez que les tables du **TP1** ont été réellement créées ? Donner toutes les informations sur ces tables ?







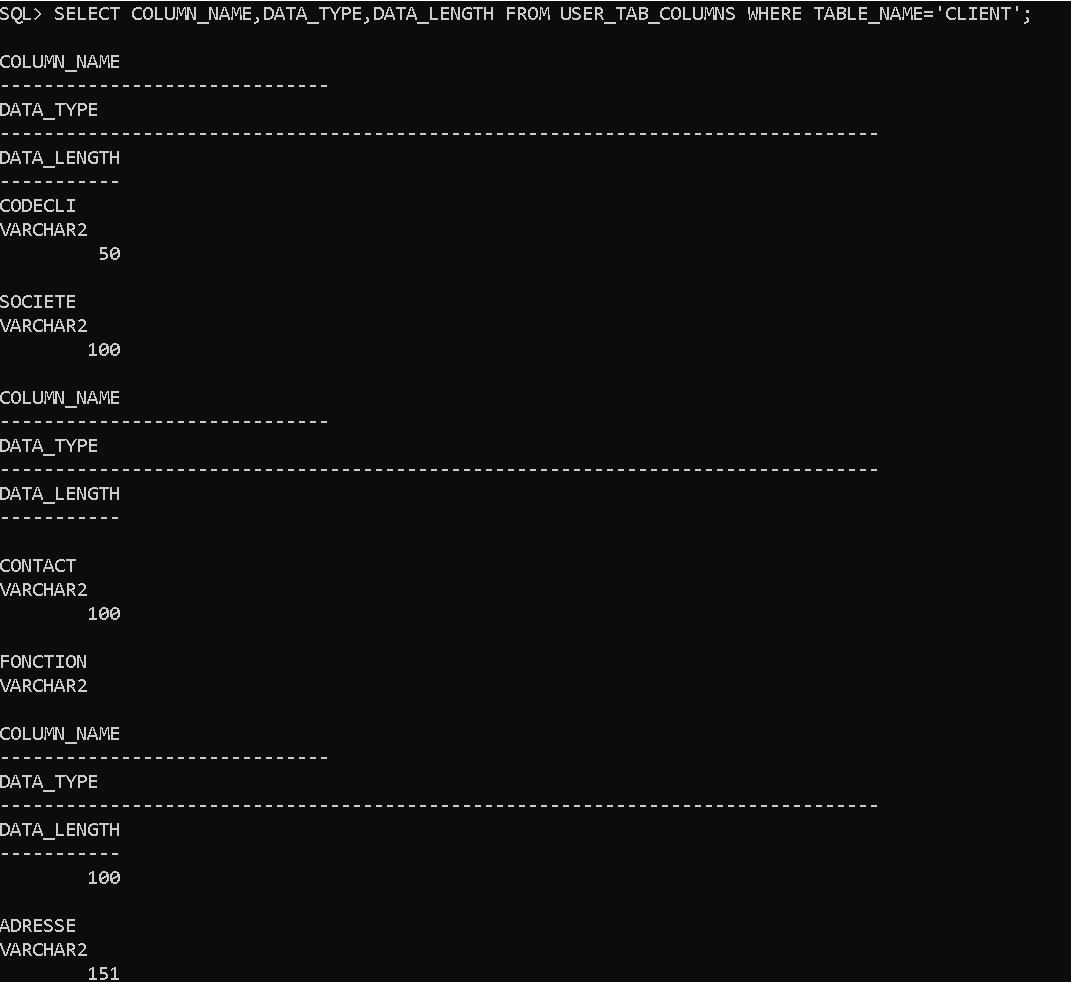
6-Lister les tables de l’utilisateur « **system** » et celles de l’utilisateur **DBACOMPTOIRS (l’utilisateur de TP1)**.





7- Donner la description des attributs des tables **Client** et **Commande** (Exploiter la table **USER\_TAB\_COLUMNS**).

Table client:



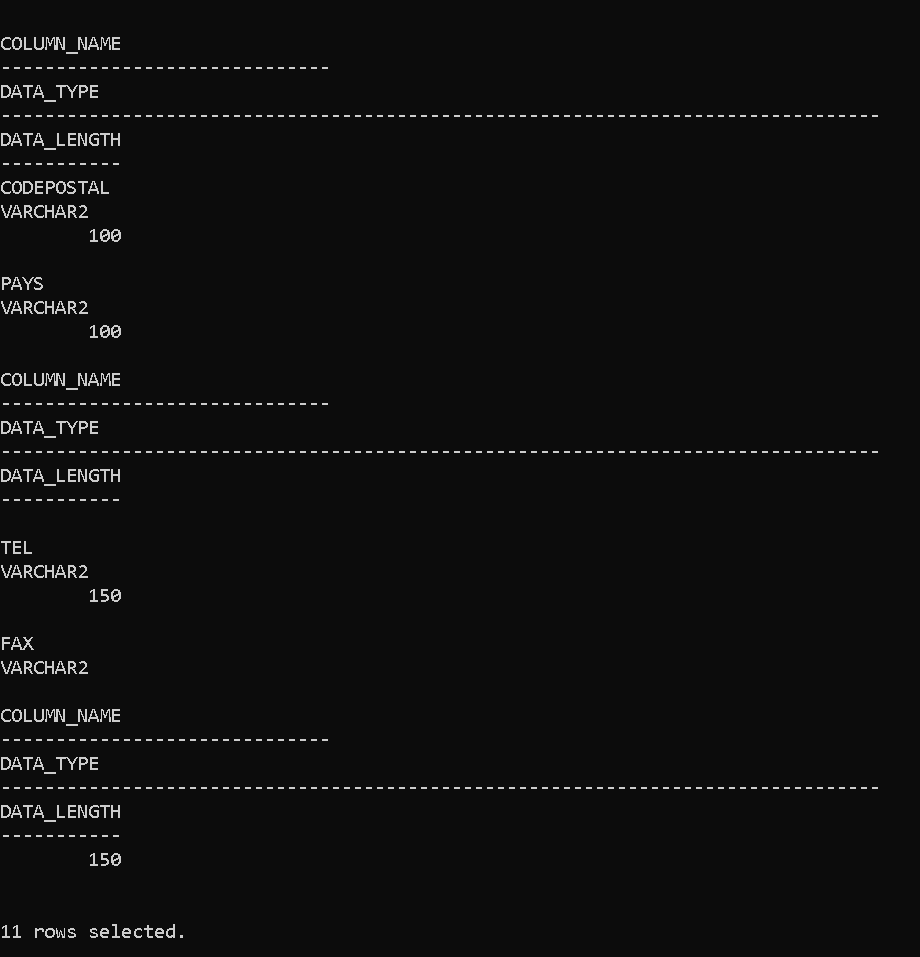
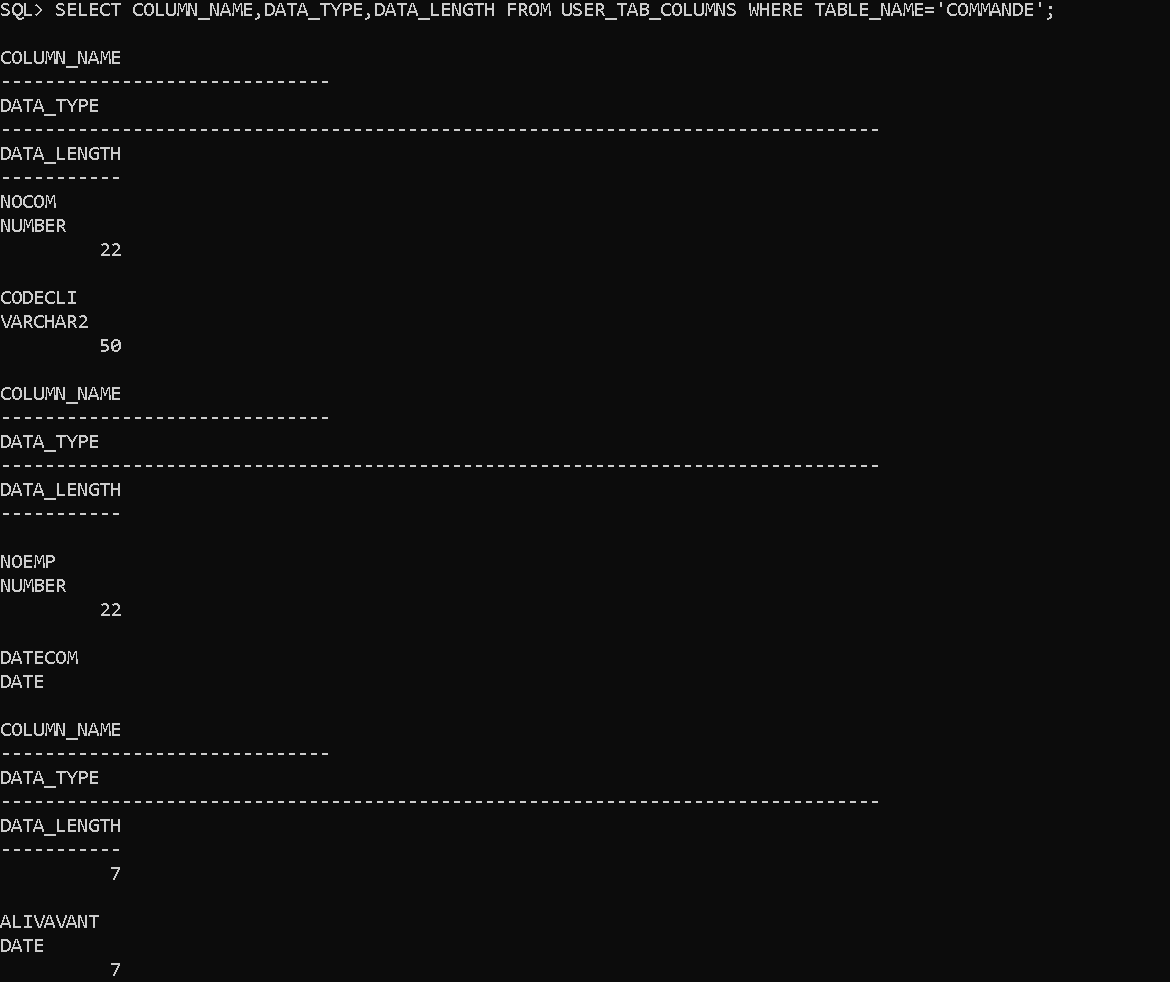
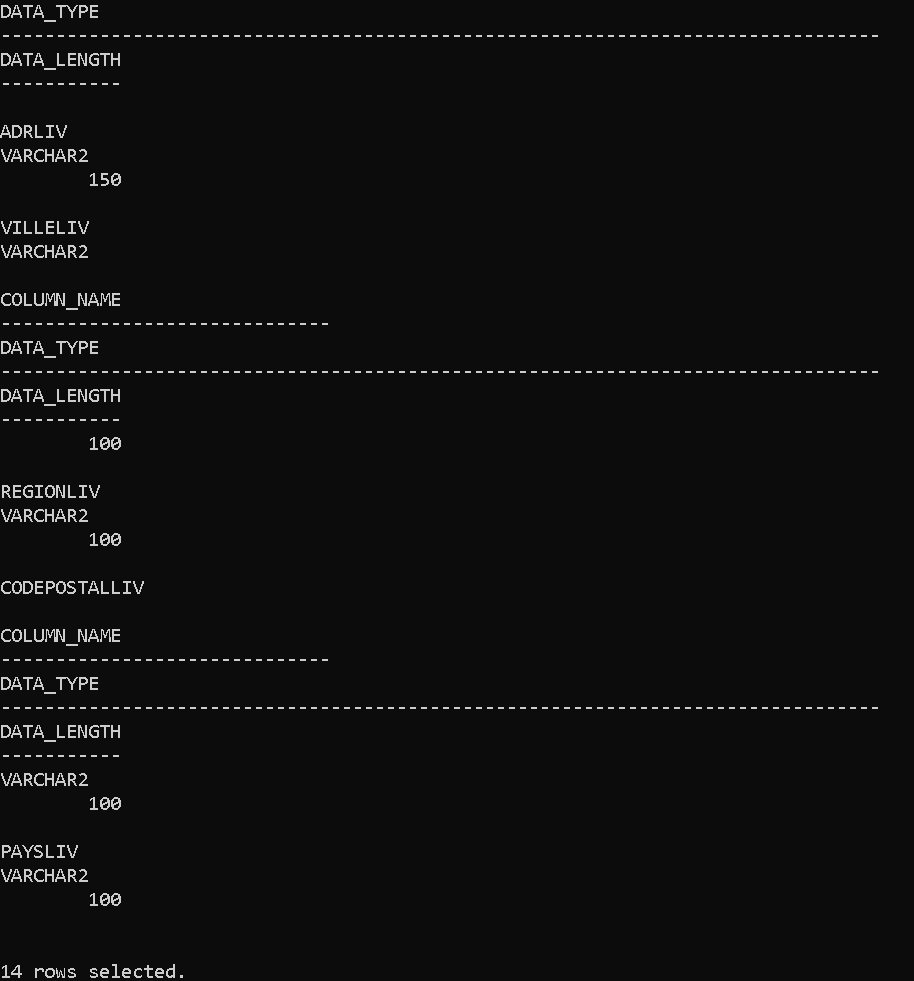


Table COMMANDE:





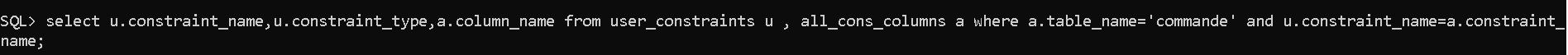
8-Comment peut-on vérifie qu’il y a une référence de clé étrangère entre les tables **CATEGORIE** et **PRODUIT**?

**On doit chercher dans le dict les tables qui contiennent le mot ALL\_CONSTRAINTS , ensuite on extrait les noms des tables qui peuvent contenir la contrainte (user\_constraints) puis on vérifie l’existence de la contrainte dans la table .**

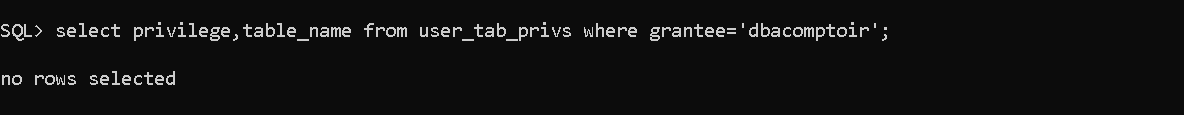
9- Donner toutes les contraintes créées lors du **TP1** et les informations qui les caractérisent (Exploitez la table **USER\_CONSTRAINTS**);



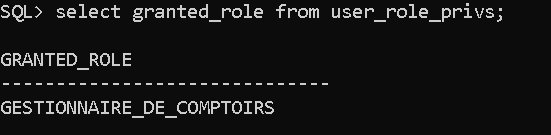
10- Retrouver toutes les informations permettant de recréer la table **COMMANDE**.



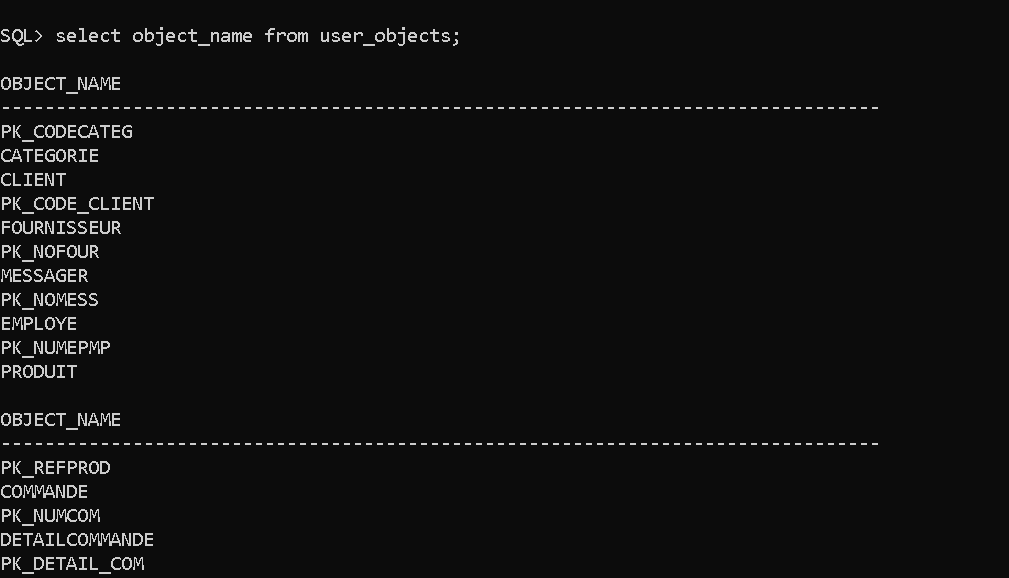
11- Trouver tous les privilèges accordés à **ADMINCOMPTOIRS.**



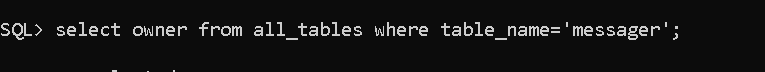
12- Trouver les rôles donnés à l’utilisateur **ADMINCOMPTOIRS**.



13- Trouver tous les objets appartenant à **ADMINCOMPTOIRS .**



14- L’administrateur cherche le propriétaire de la table **MESSAGER**, comment il pourra le trouver ?

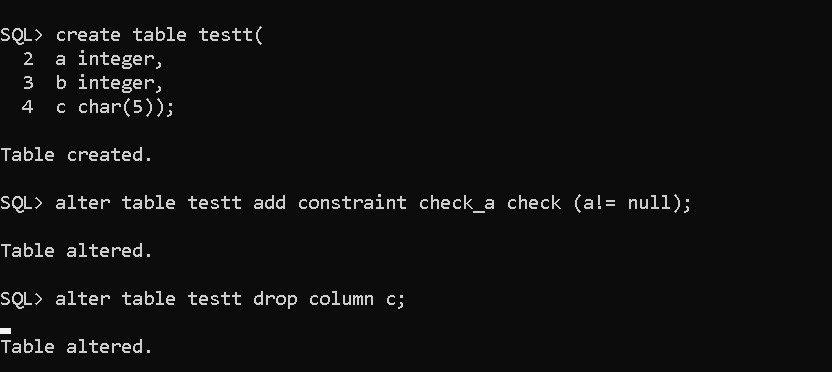


15- Donner la taille en Ko de la table **EMPLOYE**.



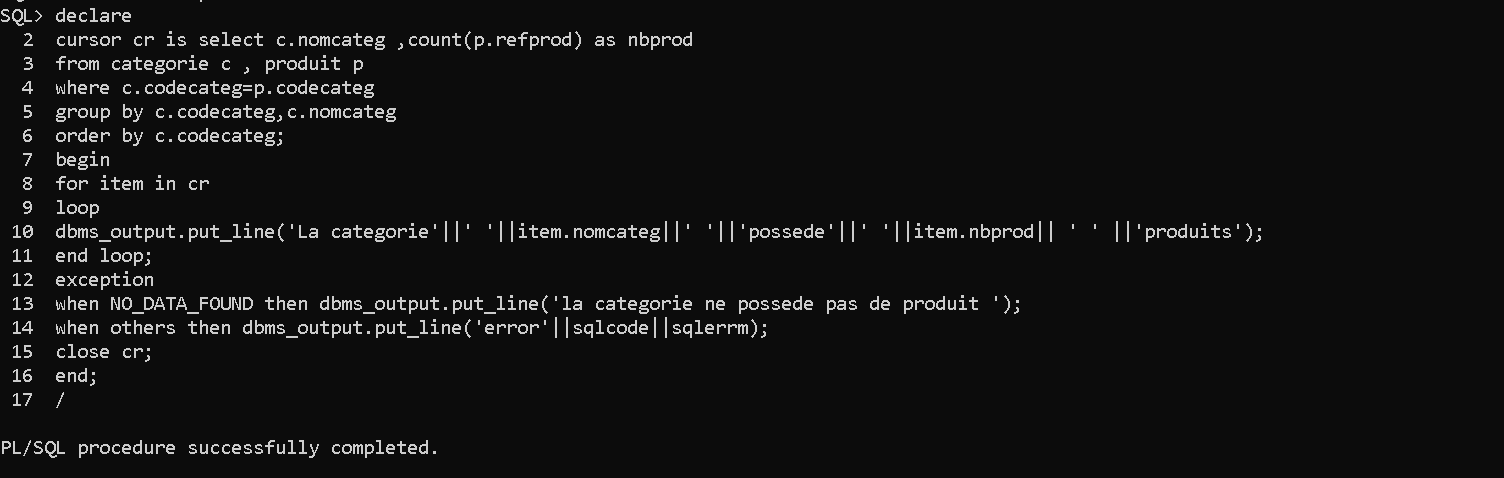
16- Vérifier l’effet produit par chacune des commandes de définition de données du **TP1** sur le dictionnaire.

On va vérifier pour la création d’une table , d’une contrainte et la suppression d’une colonne

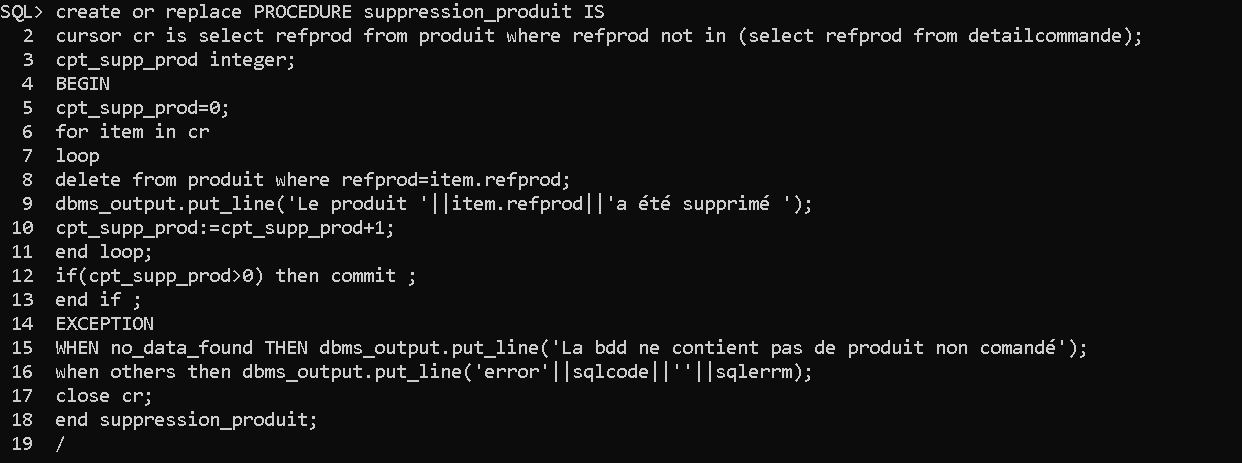


**TP N°4: PLSQL**

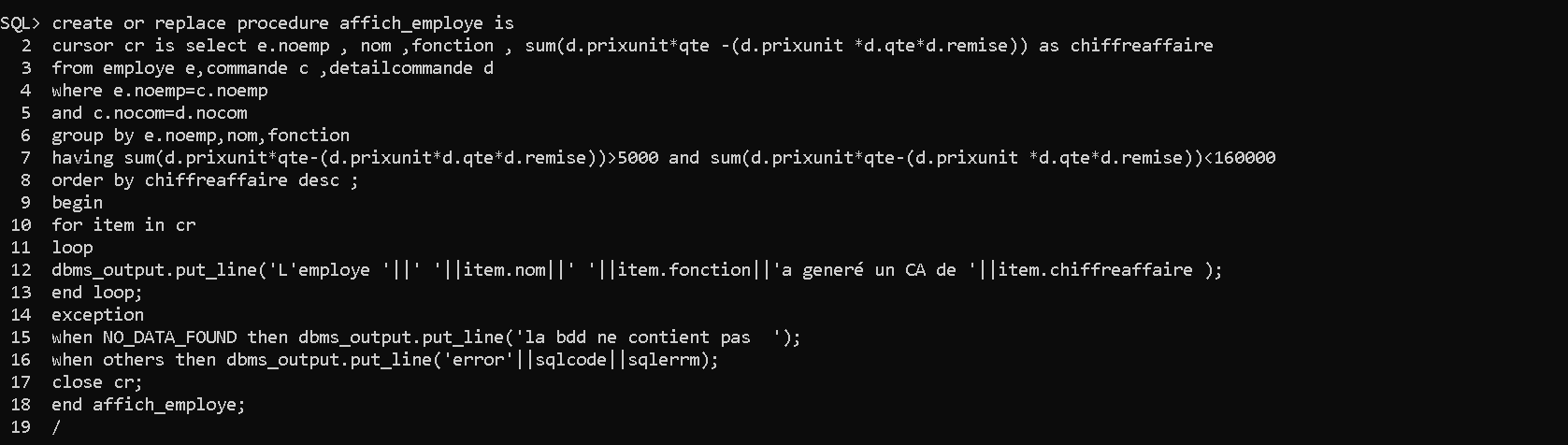
1. Ecrire un code PL/SQL qui permet d’afficher pour chaque catégorie le nombre total de ses produits.



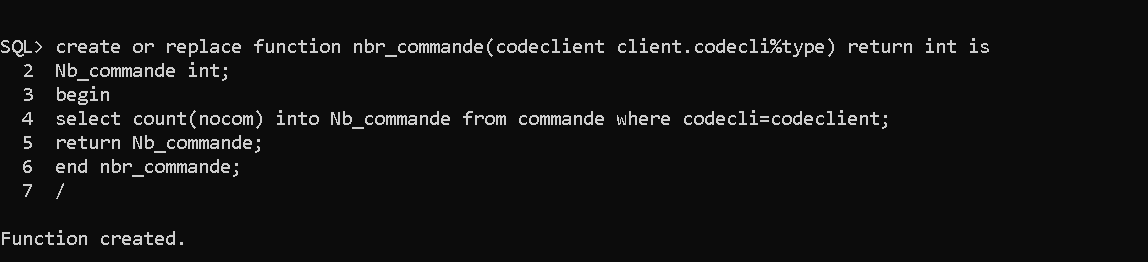
2-Ecrire une procédure qui supprime les produits qui n’étaient jamais commandés.

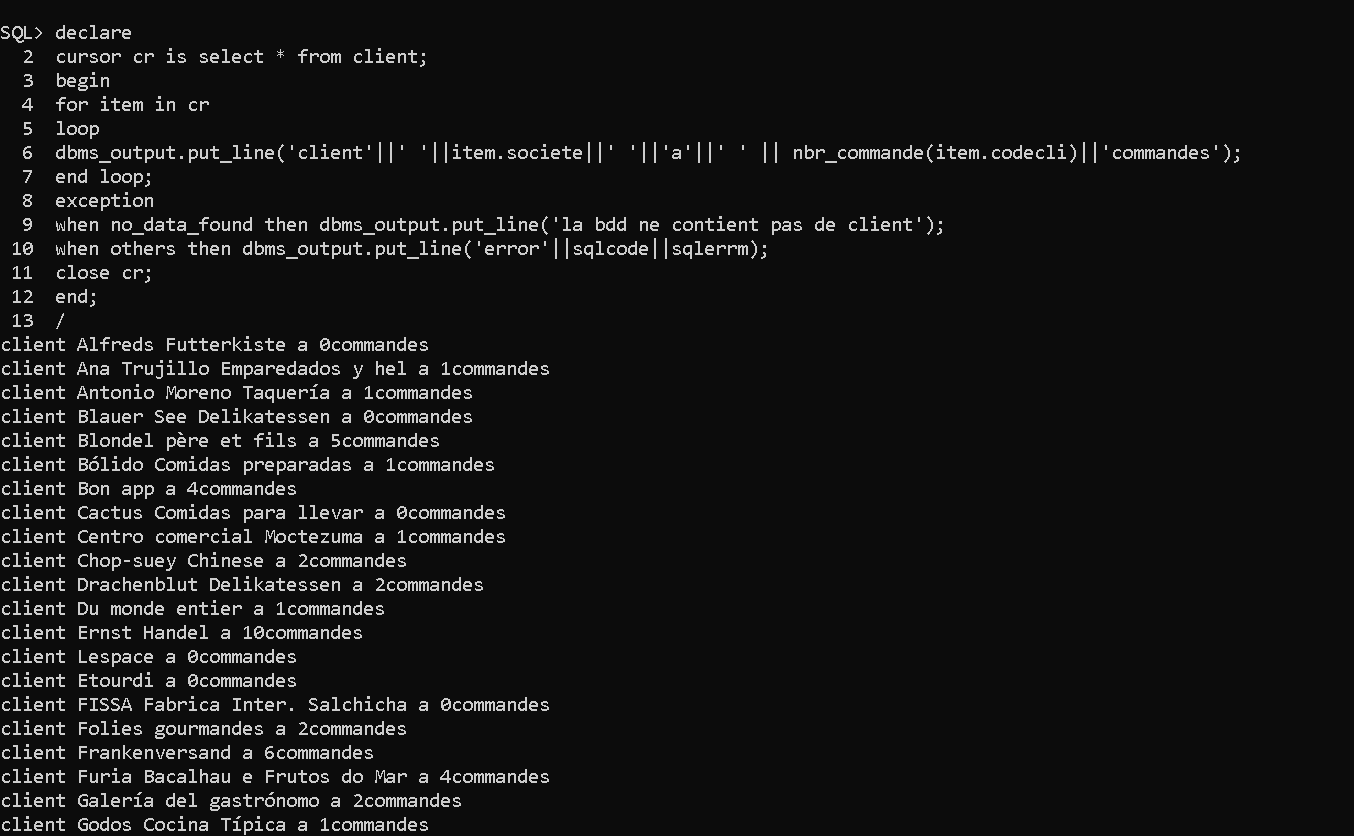


3- Ecrire une procédure qui affiche le nom, la fonction et le chiffre d'affaire généré des employés dont le chiffre d'affaire généré par ses commandes est entre 10000DA et 8000DA.

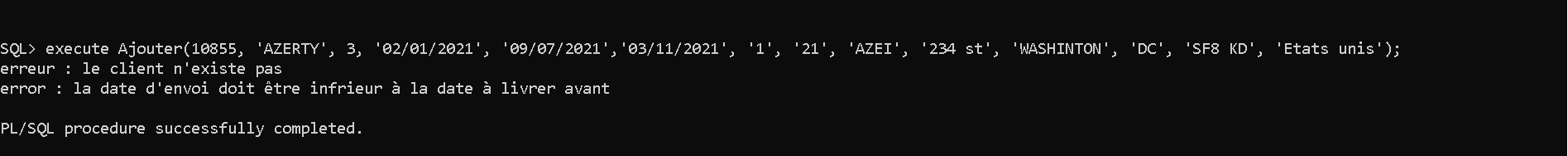
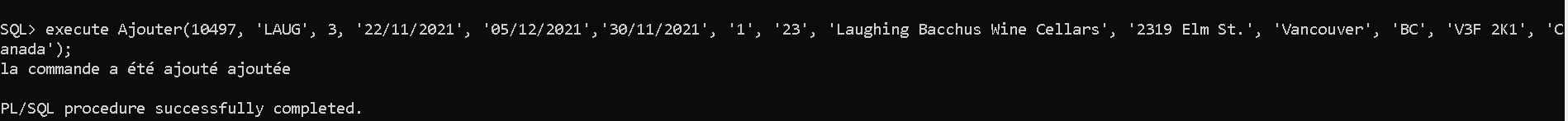
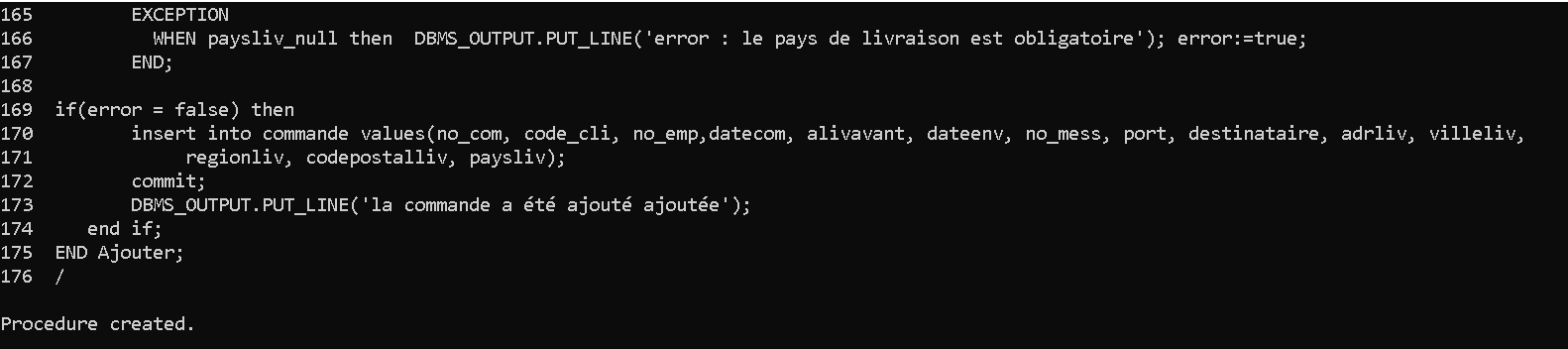
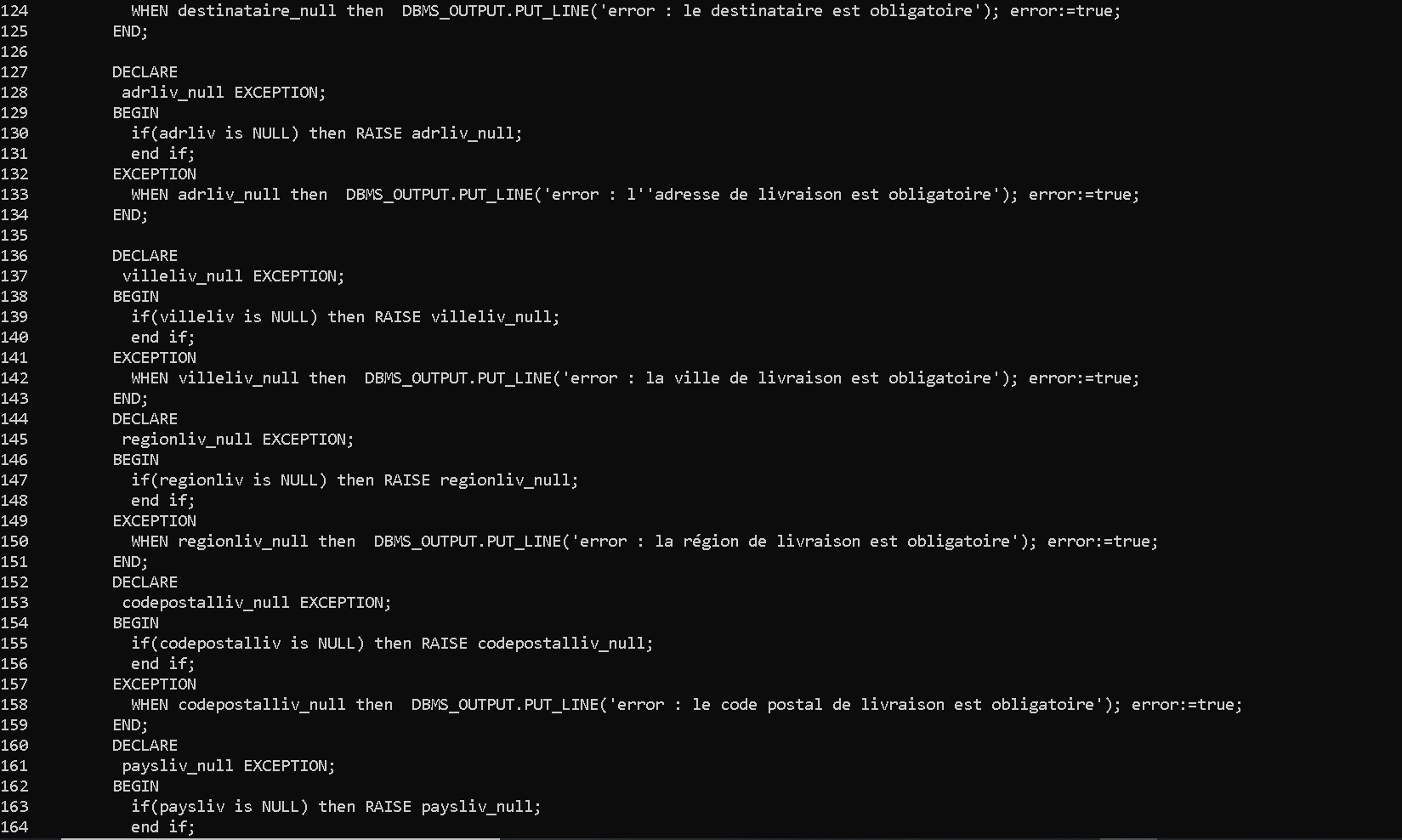
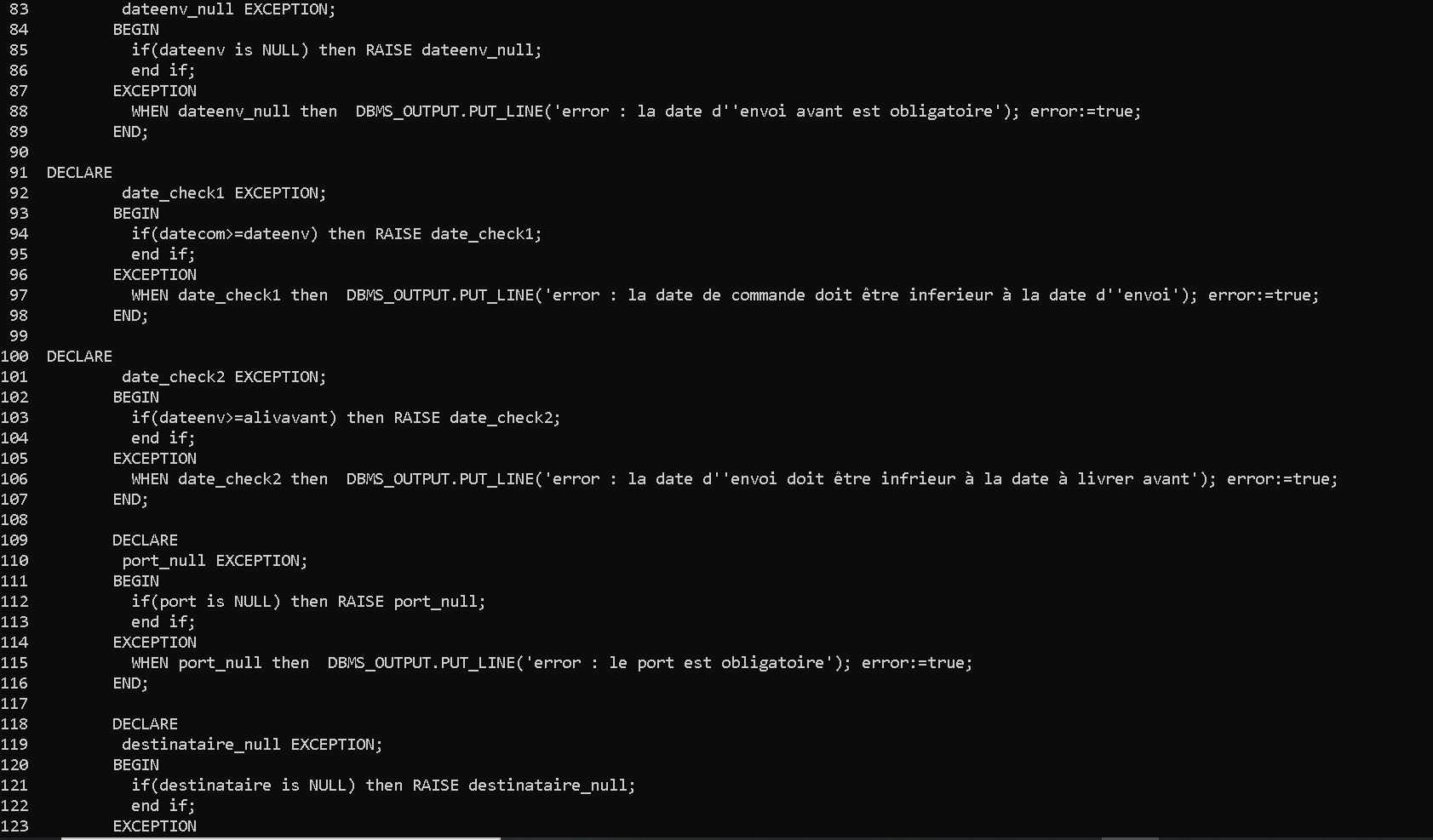
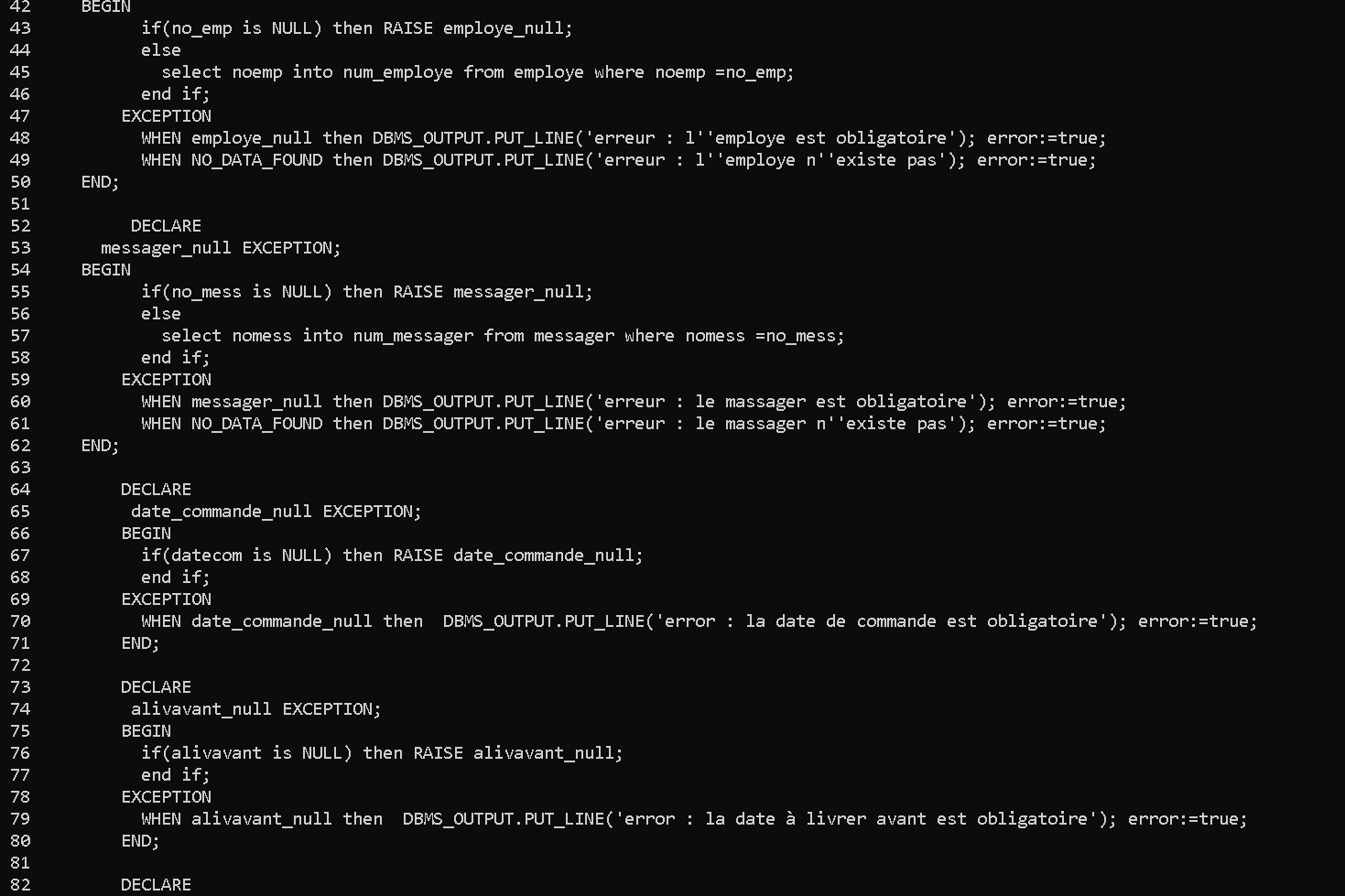
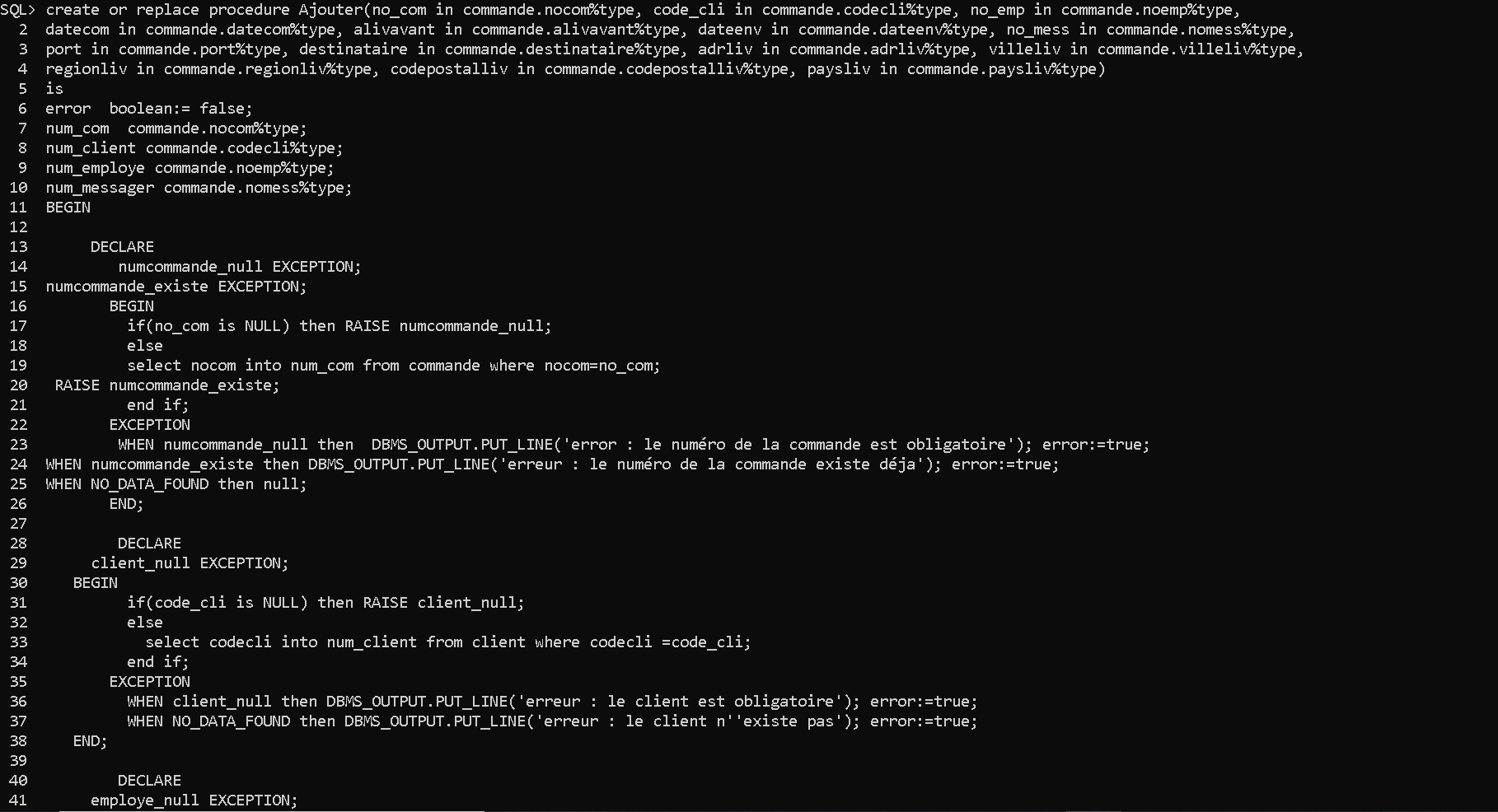


4- Ecrire une fonction qui retourne, pour chaque **client** donné, le **nombre** **de ses commandes**. Exécuter la fonction pour plusieurs clients.



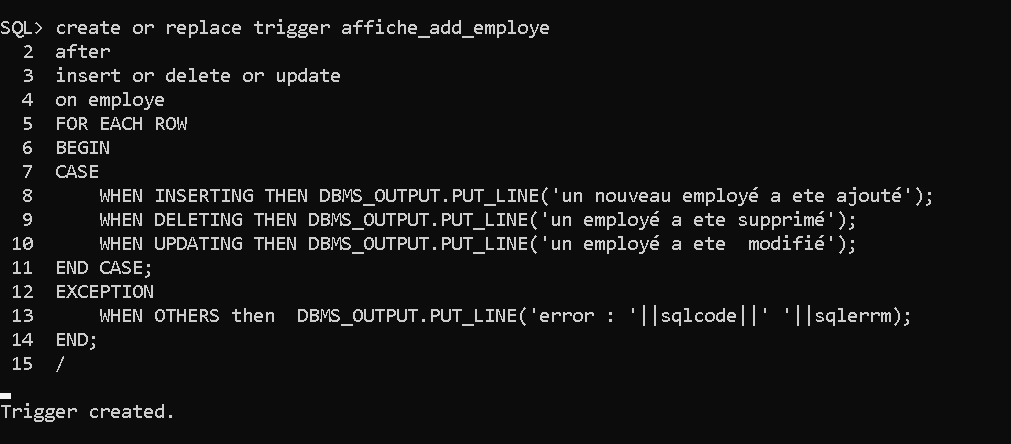


5- Créer une procédure qui permet d’ajouter **une commande** à partir de tous les attributs nécessaires. N’oublier pas de vérifier l’unicité de la clé et l’existence de clé étrangère vers **les tables référencées**. Affichez les messages d’erreurs en cas de problèmes.

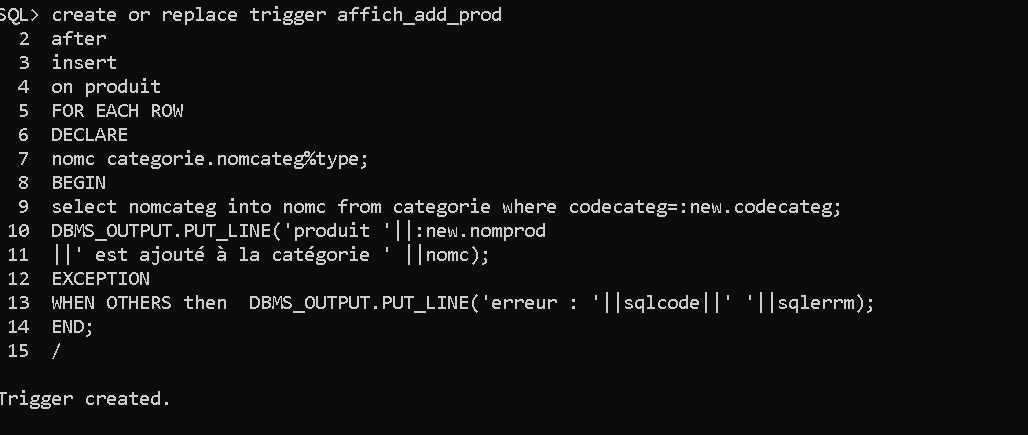


TPN°5: **TP Triggers**

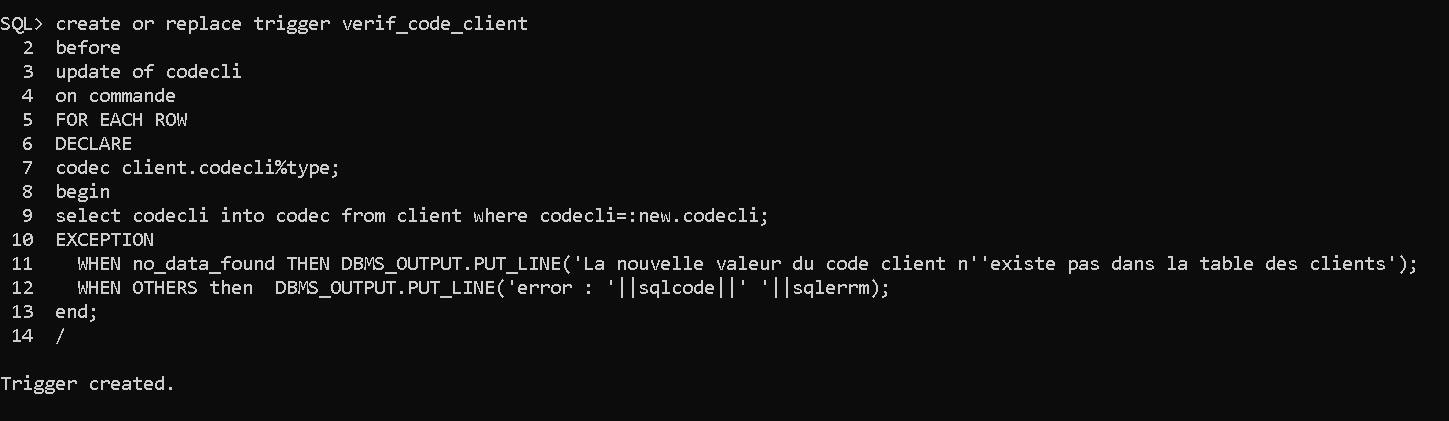
**1-** Créez un trigger qui affiche **« un nouveau employé est ajouté»** après chaque insertion d’un employé. Répétez la même chose pour la modification ou la suppression.



2- Créez un trigger qui affiche **« Un produit est ajouté à la catégorie[Nom de la catégorie] »** après chaque insertion dans la table produit.



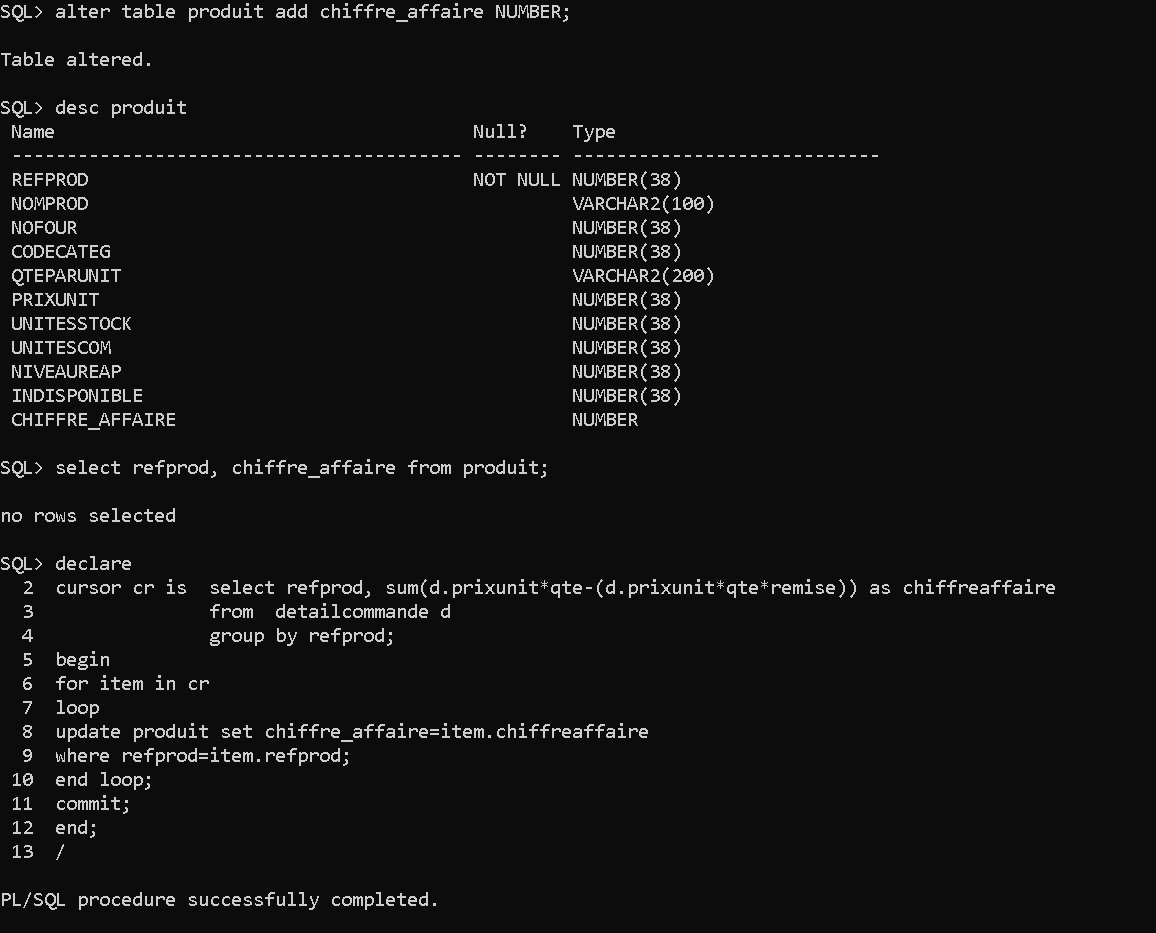
3- Créer un triggers qui vérifie avant modification du l’attribut **code client** dans la table **commande** que la nouvelle valeur existe réellement, sinon, il refuse l’opération.

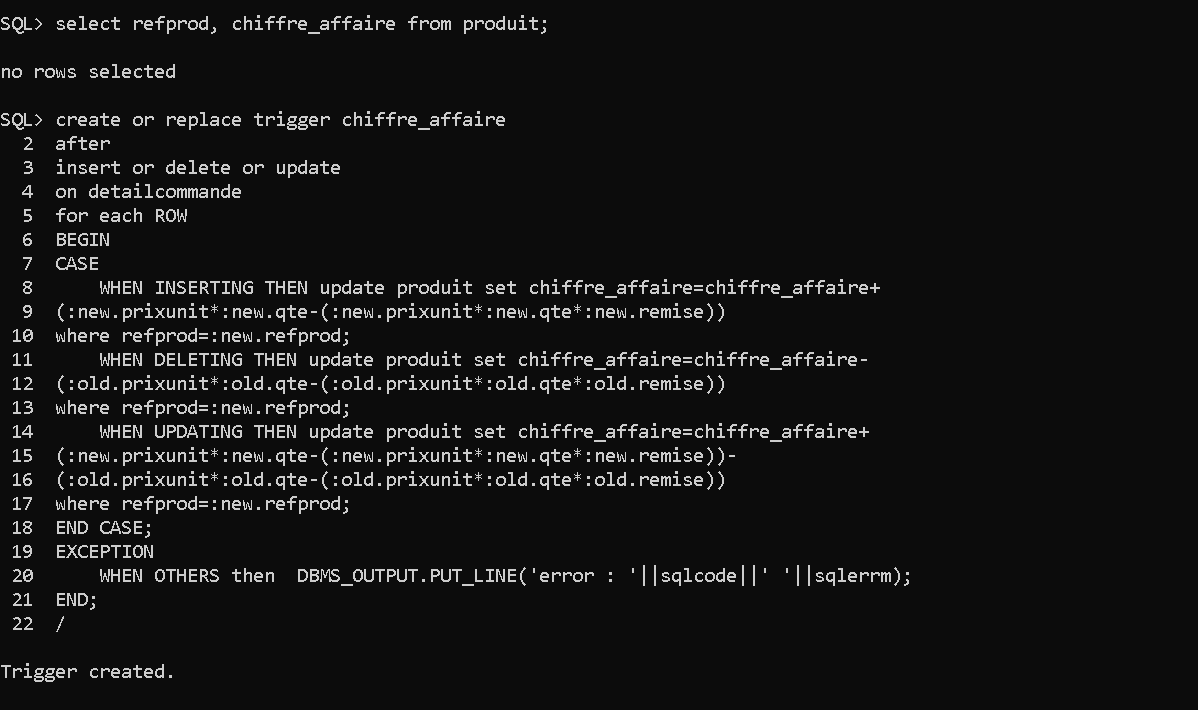


4- L’administrateur veut, pour un besoin interne, avoir le chiffre d’affaire généré par chaque **produit**. Pour cela, il ajoute un attribut : **CHIFFRE\_AFFAIRE** dans la table **produit**.

Ajouter l’attribut.

Créer un trigger **CHIFFRE\_AFFAIRE\_TRIGGER** qui mit à jour l’attribut **CHIFFRE\_AFFAIRE**lors d'une insertion, d'une modification et une suppressiondans la table **DETAILCOMMANDE**.

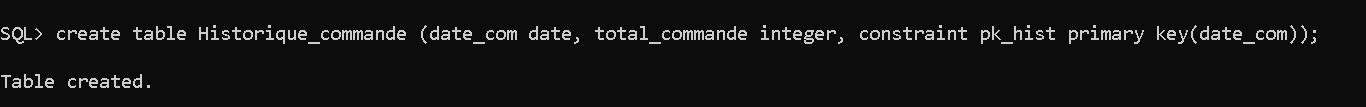
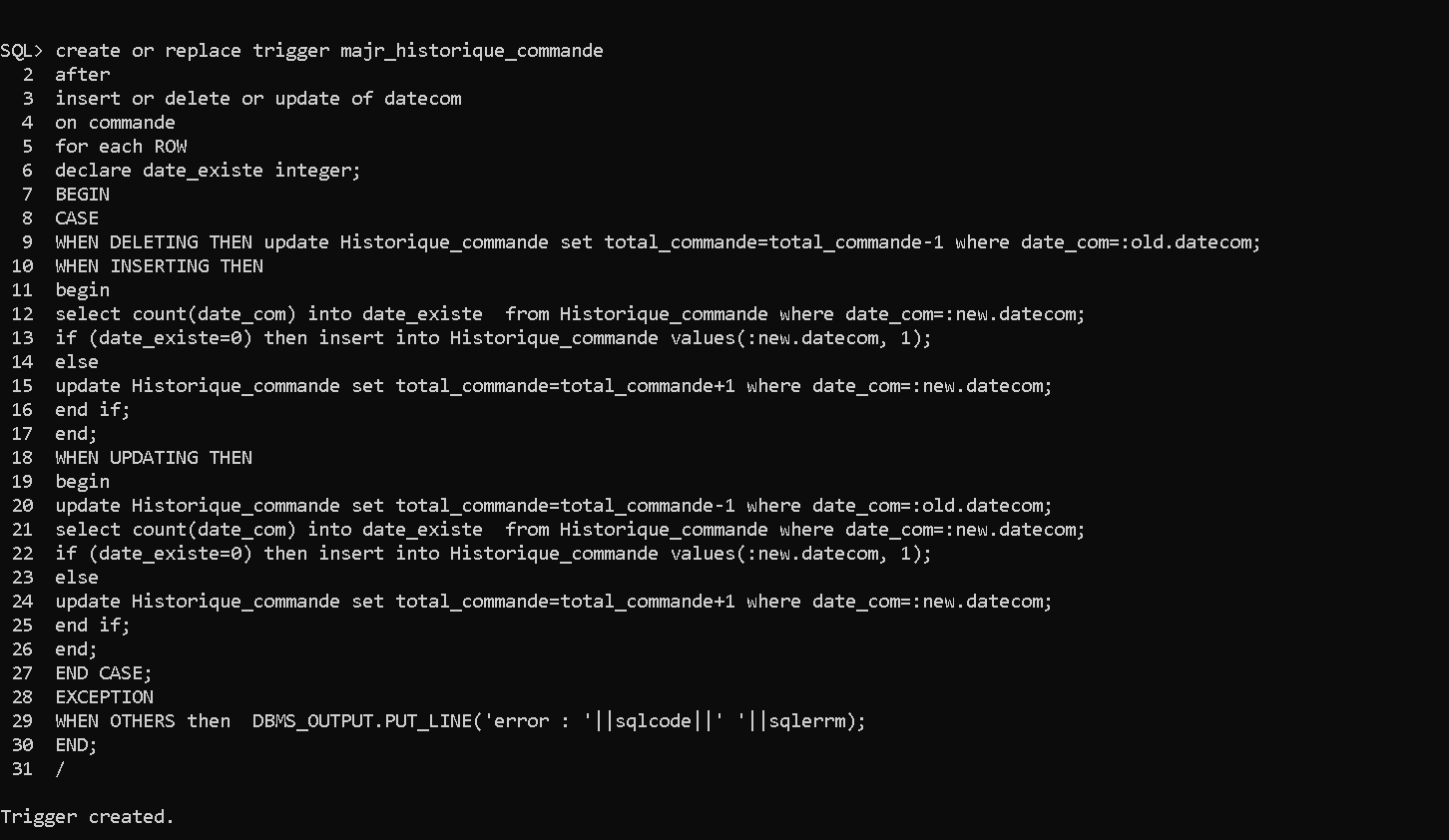




5- L’administrateur veut sauvegarder pour chaque date le nombre de commandes passées. A chaque fois une commande est ajoutée, on ajoute une ligne dans la table **Historique\_commande (DateC, total\_commande)** ou bien on mit à jour l’attribut **total\_commande**.

Ajouter la table.

Vérifier cette contrainte lors d'une insertion, d'une modification et une suppression.

Verification:

