



Administration Bases de Données Oracle

SOUTENANCE MINI-PROJET

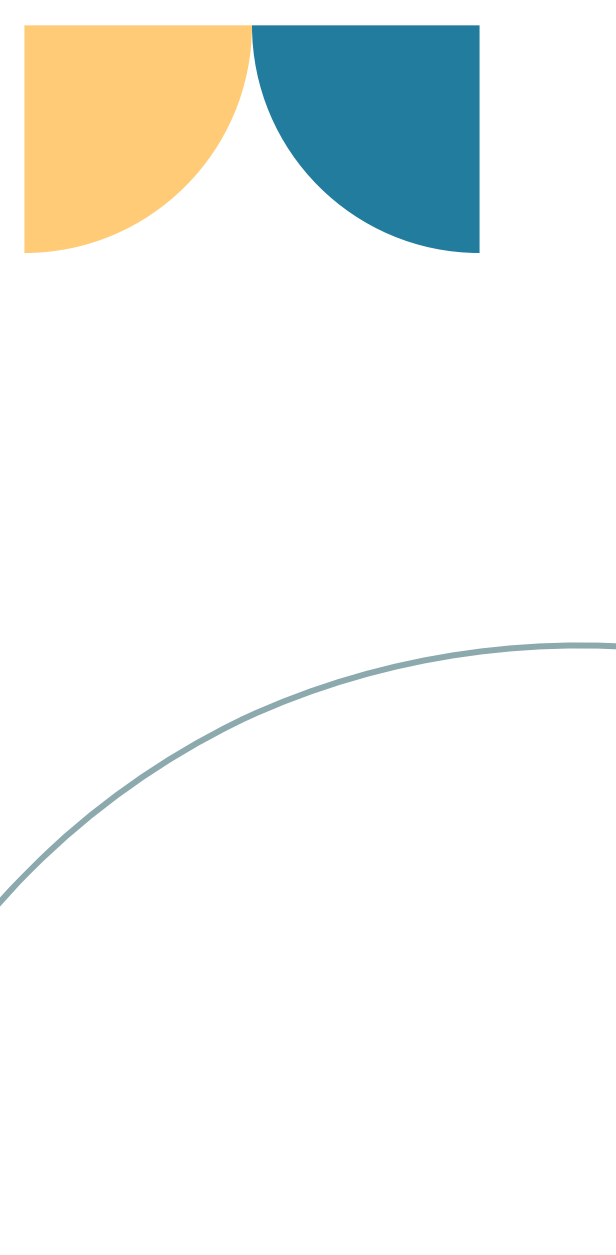


ENCADRÉ PAR

Mariam BENLLARCH

GROUPE

Osama Mansouri
Yassir Moulzim EL Alaoui
Youness Achakroun
Somaya Ayouch
Nada Jamim





PROBLEMATIQUE

Dans un projet de gestion de commandes pour une entreprise de distribution, il est crucial d'assurer la traçabilité des produits depuis le fournisseur jusqu'à la livraison chez le client final.

Le système actuel de l'entreprise est manuel, ce qui entraîne des erreurs fréquentes dans les commandes, des retards de livraison et une mauvaise gestion des stocks.

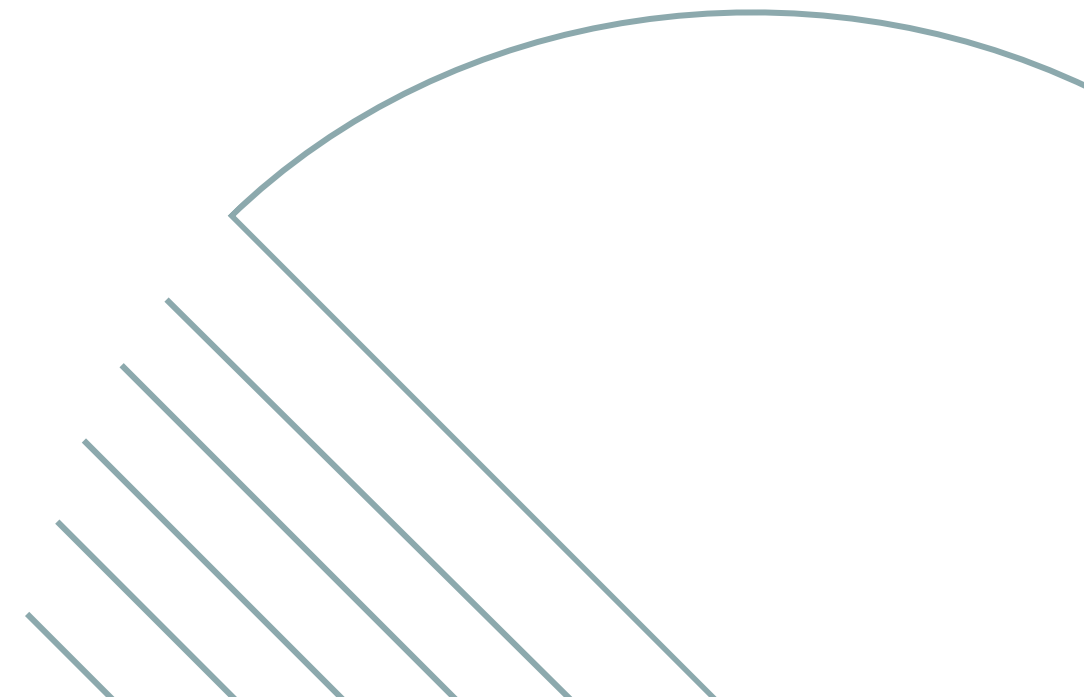
COMMENT PEUT-ON AUTOMATISER ET OPTIMISER CE PROCESSUS?



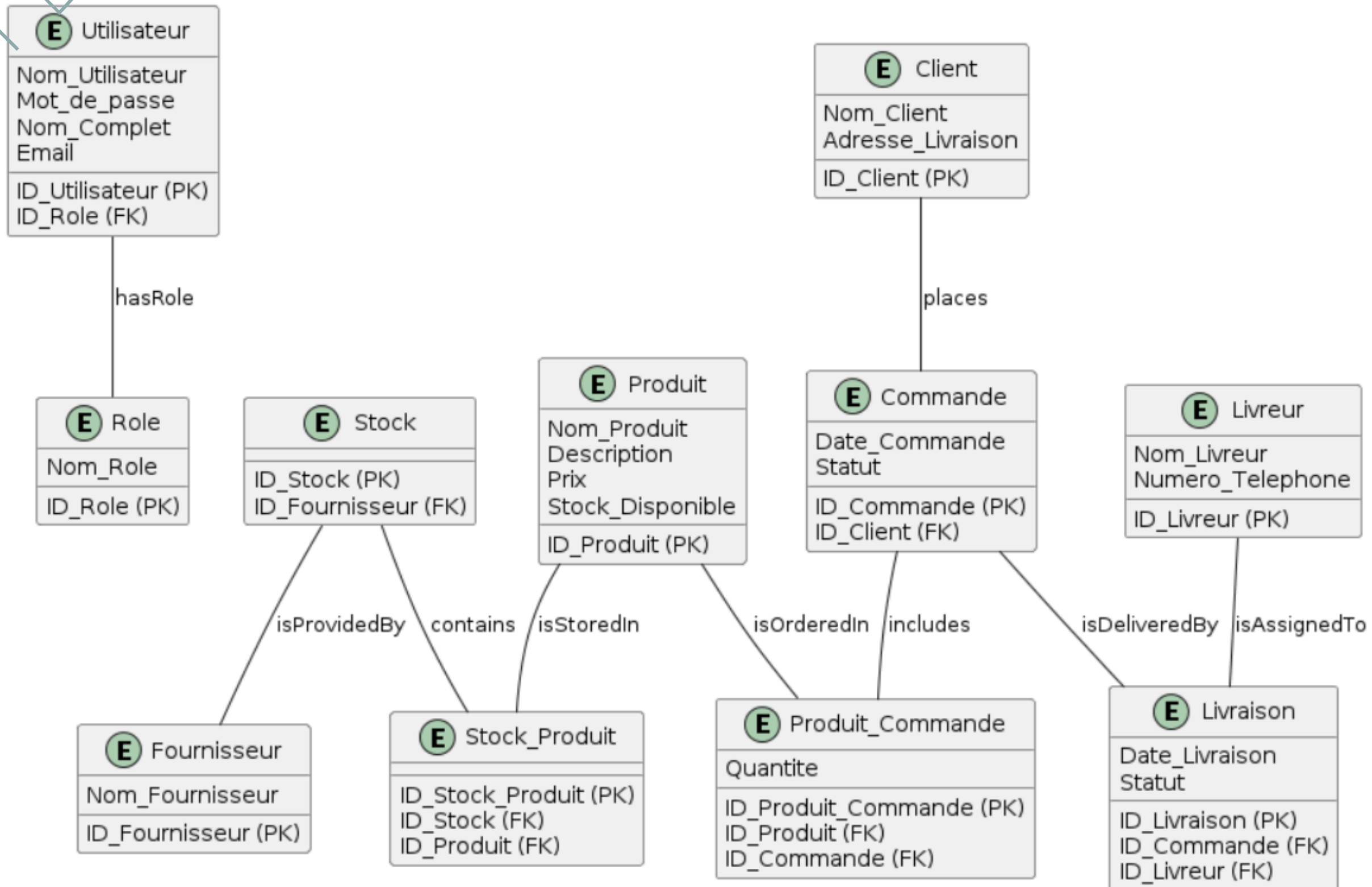
SOLUTION

Pour résoudre cette problématique, nous proposons de concevoir un système de gestion des commandes basé sur une base de données relationnelle. Voici les conceptions de la Base de données utilisé:

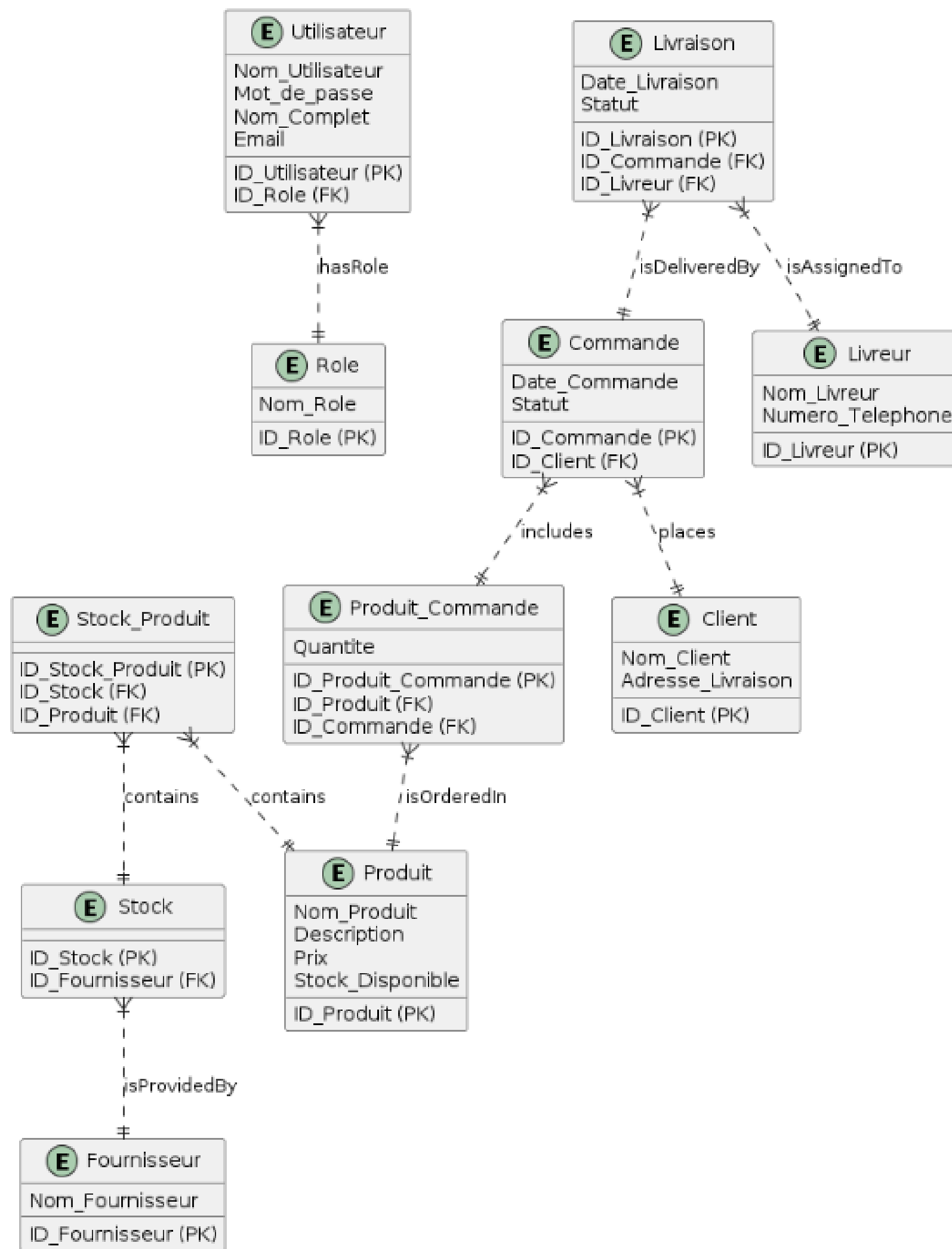
- Modèle Conceptuel des Données (MCD) : Créer un MCD pour représenter les entités principales et leurs relations.
- Modèle Logique des Données (MLD) : Traduire le MCD en MLD pour structurer la base de données de manière optimale.



MODÈLE CONCEPTUEL DES DONNÉES (MCD)



MODÈLE LOGIQUE DES DONNÉES (MLD)



LES ETAPES:

Pour créer ce projet en utilisant une base de données Oracle, nous allons suivre les étapes suivantes :

01 - CRÉER LES TABLES AVEC LES CONTRAINTES APPROPRIÉES.

02 - AJOUTER DES SÉQUENCES POUR LES CLÉS PRIMAIRES

03 - CRÉER LES INDEX POUR OPTIMISER LES REQUÊTES.

04 - INSERER LES DONNEES.

05 - CRÉER LES UTILISATEURS, LES RÔLES, ET LES PRIVILEGES

06 - TESTER LES PRIVILEGES.



01 - CRÉER LES TABLES AVEC LES CONTRAINTES APPROPRIÉES.

CREATION DES TABLES:

```
SQL> CREATE TABLE Role ( ID_Role NUMBER PRIMARY KEY, Nom_Role VARCHAR2(50) NOT NULL );
```

Table created.

```
SQL> CREATE TABLE Utilisateur ( ID_Utilisateur NUMBER PRIMARY KEY, Nom_Utilisateur VARCHAR2(50) NOT NULL, Mot_de_passe VARCHAR2(50) NOT NULL, Nom_Complet VARCHAR2(100) NOT NULL, Email VARCHAR2(100) NOT NULL, ID_Role NUMBER, CONSTRAINT fk_utilisateur_role FOREIGN KEY (ID_Role) REFERENCES Role(ID_Role) );
```

Table created.

```
SQL> CREATE TABLE Fournisseur ( ID_Fournisseur NUMBER PRIMARY KEY, Nom_Fournisseur VARCHAR2(100) NOT NULL );
```

Table created.

```
SQL> CREATE TABLE Stock ( ID_Stock NUMBER PRIMARY KEY, ID_Fournisseur NUMBER, CONSTRAINT fk_stock_fournisseur FOREIGN KEY (ID_Fournisseur) REFERENCES Fournisseur(ID_Fournisseur) );
```

Table created.

```
SQL> CREATE TABLE Stock ( ID_Stock NUMBER PRIMARY KEY, ID_Fournisseur NUMBER, CONSTRAINT fk_stock_fournisseur FOREIGN KEY (ID_Fournisseur) REFERENCES Fournisseur(ID_Fournisseur) );
```

```
CREATE TABLE Stock ( ID_Stock NUMBER PRIMARY KEY, ID_Fournisseur NUMBER, CONSTRAINT fk_stock_fournisseur FOREIGN KEY (ID_Fournisseur) REFERENCES Fournisseur(ID_Fournisseur) )
```

*

ERROR at line 1:

ORA-00955: name is already used by an existing object

```
SQL> CREATE TABLE Produit ( ID_Produit NUMBER PRIMARY KEY, Nom_Produit VARCHAR2(100) NOT NULL, Description VARCHAR2(500), Prix NUMBER(10, 2) NOT NULL, Stock_Disponible NUMBER );
```

Table created.


```
SQL> CREATE TABLE Stock_Produit (  
2     ID_Stock_Produit NUMBER PRIMARY KEY,  
3     ID_Stock NUMBER,  
4     ID_Produit NUMBER,  
5     CONSTRAINT fk_stock_produit_stock FOREIGN KEY (ID_Stock) REFERENCES Stock(ID_Stock),  
6     CONSTRAINT fk_stock_produit_produit FOREIGN KEY (ID_Produit) REFERENCES Produit(ID_Produit)  
7 );
```

Table created.

```
SQL> CREATE TABLE Client (  
2     ID_Client NUMBER PRIMARY KEY,  
3     Nom_Client VARCHAR2(100) NOT NULL,  
4     Adresse_Livraison VARCHAR2(200) NOT NULL  
5 );
```

Table created.

```
SQL> CREATE TABLE Commande (  
2     ID_Commande NUMBER PRIMARY KEY,  
3     Date_Commande DATE NOT NULL,  
4     Statut VARCHAR2(50) NOT NULL,  
5     ID_Client NUMBER,  
6     CONSTRAINT fk_commande_client FOREIGN KEY (ID_Client) REFERENCES Client(ID_Client)  
7 );
```

Table created.

```
SQL> CREATE TABLE Produit_Commande (  
2     ID_Produit_Commande NUMBER PRIMARY KEY,  
3     ID_Produit NUMBER,  
4     ID_Commande NUMBER,  
5     Quantite NUMBER NOT NULL,  
6     CONSTRAINT fk_produit_commande_produit FOREIGN KEY (ID_Produit) REFERENCES Produit(ID_Produit),  
7     CONSTRAINT fk_produit_commande_commande FOREIGN KEY (ID_Commande) REFERENCES Commande(ID_Commande)  
8 );
```

Table created.



VERIFICATION DES TABLES INSÉRÉS:

```
SQL> SELECT table_name FROM user_tables;
```

```
TABLE_NAME
```

```
-----
```

```
ROLE
```

```
UTILISATEUR
```

```
FOURNISSEUR
```

```
STOCK
```

```
PRODUIT
```

```
STOCK_PRODUIT
```

```
CLIENT
```

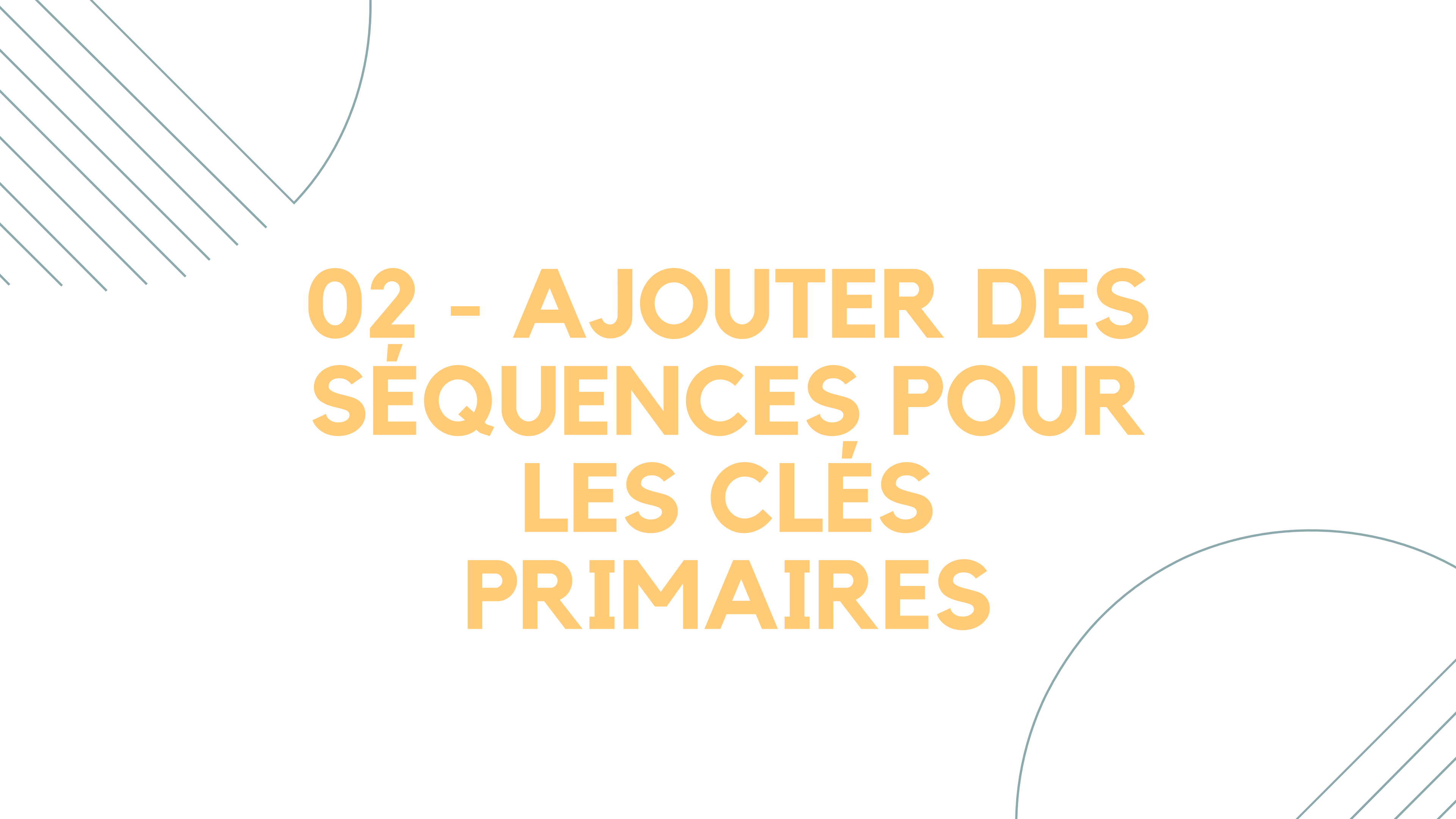
```
COMMANDE
```

```
PRODUIT_COMMANDE
```

```
LIVREUR
```

```
LIVRAISON
```

```
11 rows selected.
```



02 - AJOUTER DES SÉQUENCES POUR LES CLÉS PRIMAIRES

CREATION DES SEQUENCES:

```
SQL> CREATE SEQUENCE seq_role START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_utilisateur START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_fournisseur START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_stock START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_produit START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_stock_produit START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_client START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_commande START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_produit_commande START WITH 1 INCREMENT BY 1;
Sequence created.

SQL> CREATE SEQUENCE seq_livreur START WITH 1 INCREMENT BY 1;
Sequence created.
```

VERIFICATION DES SEQUENCES INSÉRÉS:

```
SQL> SELECT sequence_name, last_number FROM user_sequences;
```

SEQUENCE_NAME	LAST_NUMBER
-----	-----
SEQ_CLIENT	21
SEQ_COMMANDE	21
SEQ_FOURNISSEUR	21
SEQ_LIVRAISON	21
SEQ_LIVREUR	21
SEQ_PRODUIT	21
SEQ_PRODUIT_COMMANDE	21
SEQ_ROLE	21
SEQ_STOCK	21
SEQ_STOCK_PRODUIT	21
SEQ_UTILISATEUR	21

```
11 rows selected.
```





03 - CRÉER LES INDEX POUR OPTIMISER LES REQUÊTES.

```
SQL> CREATE INDEX idx_utilisateur_role ON Utilisateur(ID_Role);  
Index created.  
  
SQL> CREATE INDEX idx_stock_fournisseur ON Stock(ID_Fournisseur);  
Index created.  
  
SQL> CREATE INDEX idx_stock_produit_stock ON Stock_Produit(ID_Stock);  
Index created.  
  
SQL> CREATE INDEX idx_stock_produit_produit ON Stock_Produit(ID_Produit);  
Index created.  
  
SQL> CREATE INDEX idx_commande_client ON Commande(ID_Client);  
Index created.  
  
SQL> CREATE INDEX idx_produit_commande_produit ON Produit_Commande(ID_Produit);  
Index created.  
  
SQL> CREATE INDEX idx_produit_commande_commande ON Produit_Commande(ID_Commande);  
Index created.  
  
SQL> CREATE INDEX idx_livraison_commande ON Livraison(ID_Commande);  
Index created.  
  
SQL> CREATE INDEX idx_livraison_livreur ON Livraison(ID_Livreur);  
Index created.
```



CREATION DES INDEXS:

VERIFICATION DES INDEXS INSÉRÉS:



```
SQL> SELECT index_name, table_name FROM user_indexes;
```

INDEX_NAME	TABLE_NAME
SYS_C007092	UTILISATEUR
IDX_UTILISATEUR_ROLE	UTILISATEUR
SYS_C007101	STOCK_PRODUIT
IDX_STOCK_PRODUIT_STOCK	STOCK_PRODUIT
IDX_STOCK_PRODUIT_PRODUIT	STOCK_PRODUIT
SYS_C007096	STOCK
IDX_STOCK_FOURNISSEUR	STOCK
SYS_C007087	ROLE
SYS_C007112	PRODUIT_COMMANDE
IDX_PRODUIT_COMMANDE_PRODUIT	PRODUIT_COMMANDE
IDX_PRODUIT_COMMANDE_COMMANDE	PRODUIT_COMMANDE

INDEX_NAME	TABLE_NAME
SYS_C007100	PRODUIT
SYS_C007117	LIVREUR
SYS_C007120	LIVRAISON
IDX_LIVRAISON_COMMANDE	LIVRAISON
IDX_LIVRAISON_LIVREUR	LIVRAISON
SYS_C007095	FOURNISSEUR
SYS_C007109	COMMANDE
IDX_COMMANDE_CLIENT	COMMANDE
SYS_C007106	CLIENT

```
20 rows selected.
```




04 - INSERER LES DONNEES.

INSERTION DES DONNEES:

```
SQL> INSERT INTO Role (ID_Role, Nom_Role) VALUES (seq_role.NEXTVAL, 'Admin');
1 row created.

SQL> INSERT INTO Role (ID_Role, Nom_Role) VALUES (seq_role.NEXTVAL, 'Client');
1 row created.

SQL> INSERT INTO Role (ID_Role, Nom_Role) VALUES (seq_role.NEXTVAL, 'Livreur');
1 row created.

SQL> INSERT INTO Utilisateur (ID_Utilisateur, Nom_Utilisateur, Mot_de_passe, Nom_Complet, Email, ID_Role)
  2  VALUES (seq_utilisateur.NEXTVAL, 'admin_user', 'password_hash', 'Admin User', 'admin@example.com', 1);
1 row created.

SQL> INSERT INTO Fournisseur (ID_Fournisseur, Nom_Fournisseur) VALUES (seq_fournisseur.NEXTVAL, 'Fournisseur A');
1 row created.

SQL> INSERT INTO Stock (ID_Stock, ID_Fournisseur) VALUES (seq_stock.NEXTVAL, 1);
1 row created.

SQL> INSERT INTO Produit (ID_Produit, Nom_Produit, Description, Prix, Stock_Disponible)
  2  VALUES (seq_produit.NEXTVAL, 'Produit A', 'Description du Produit A', 100, 50);
1 row created.

SQL> INSERT INTO Stock_Produit (ID_Stock_Produit, ID_Stock, ID_Produit) VALUES (seq_stock_produit.NEXTVAL, 1, 1);
1 row created.

SQL> INSERT INTO Client (ID_Client, Nom_Client, Adresse_Livraison) VALUES (seq_client.NEXTVAL, 'Client A', 'Adresse A');
1 row created.
```

```
SQL> INSERT INTO Commande (ID_Commande, Date_Commande, Statut, ID_Client) VALUES (seq_commande.NEXTVAL, SYSDATE, 'En cours', 1);
1 row created.

SQL> INSERT INTO Produit_Commande (ID_Produit_Commande, ID_Produit, ID_Commande, Quantite) VALUES (seq_produit_commande.NEXTVAL, 1, 1, 2);
1 row created.

SQL> INSERT INTO Livreur (ID_Livreur, Nom_Livreur, Numero_Telephone) VALUES (seq_livreur.NEXTVAL, 'Livreur A', '123456789');
1 row created.

SQL> INSERT INTO Livraison (ID_Livraison, Date_Livraison, Statut, ID_Commande, ID_Livreur) VALUES (seq_livraison.NEXTVAL, SYSDATE + 1, 'En cours', 1, 1);
1 row created.
```



VERIFICATION DES DONNEES INSÉRÉS:

```
SQL> SELECT ID_Utilisateur, Nom_Utilisateur, Mot_de_passe, Nom_Complet, Email, ID_Role FROM Utilisateur;
```

ID_UTILISATEUR	NOM_UTILISATEUR	MOT_DE_PASSE	NOM_COMPLET	EMAIL	ID_ROLE
1	admin_user	password_hash	Admin User	admin@example.com	1

```
SQL> SELECT * FROM Fournisseur;
```

ID_FOURNISSEUR	NOM_FOURNISSEUR
1	Fournisseur A

```
SQL> SELECT * FROM Role;
```

ID_ROLE	NOM_ROLE
1	Admin
2	Client
3	Livreur

```
SQL> SELECT * FROM Stock;
```

ID_STOCK	ID_FOURNISSEUR
1	1

```
SQL> SELECT * FROM Produit;
```

rows will be truncated

rows will be truncated

ID_PRODUIT	NOM_PRODUIT	DESCRIPTION
1	Produit A	Description du Produit A

```
SQL> SELECT * FROM Stock_Produit;
```

ID_STOCK_PRODUIT	ID_STOCK	ID_PRODUIT
1	1	1

```
SQL> SELECT * FROM Client;
```

ID_CLIENT	NOM_CLIENT	ADRESSE_LIVRAISON
1	Client A	Adresse A

```
SQL> SELECT * FROM Commande;
```

ID_COMMANDE	DATE_COM	STATUT	ID_CLIENT
1	24-06-08	En cours	1

```
SQL> SELECT * FROM Produit_Commande;
```

ID_PRODUIT_COMMANDE	ID_PRODUIT	ID_COMMANDE	QUANTITE
1	1	1	2

```
SQL> SELECT * FROM Livreur;
```

ID_LIVREUR	NOM_LIVREUR	NUMERO_TELEPHONE
1	Livreur A	123456789

```
SQL> SELECT * FROM Livraison;
```

ID_LIVRAISON	DATE_LIV	STATUT	ID_COMMANDE	ID_LIVREUR
1	24-06-09	En cours	1	1



05 - CRÉER LES UTILISATEURS, LES RÔLES, ET LES PRIVILEGES

CRÉER LES UTILISATEURS:

```
C:\Users\Dark>sqlplus mahalcomtest/0000@localhost/xe

SQL*Plus: Release 11.2.0.2.0 Production on Sat Jun 8 15:51:36 2024

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE USER admin_user IDENTIFIED BY admin_password;

User created.

SQL> CREATE USER client_user IDENTIFIED BY client_password;

User created.

SQL> CREATE USER livreur_user IDENTIFIED BY livreur_password;

User created.
```

CRÉER LES ROLES:

```
C:\Users\Dark>sqlplus mahalcomtest/0000@localhost/xe

SQL*Plus: Release 11.2.0.2.0 Production on Sat Jun 8 15:46:28 2024

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE ROLE client_role;

Role created.

SQL> CREATE ROLE client_role;
CREATE ROLE client_role
          *
ERROR at line 1:
ORA-01921: role name 'CLIENT_ROLE' conflicts with another user or role name

SQL> CREATE ROLE livreur_role;

Role created.
```



AFFECTER UN ROLE À UN UTILISATEUR:

```
SQL> GRANT admin_role TO admin_user;
```

```
Grant succeeded.
```

```
SQL> GRANT client_role TO client_user;
```

```
Grant succeeded.
```

```
SQL> GRANT livreur_role TO livreur_user;
```

```
Grant succeeded.
```



CRÉER LES PRIVILEGES:

```
SQL> GRANT CREATE SESSION TO admin_role;
Grant succeeded.

SQL> GRANT CREATE TABLE TO admin_role;
Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Role TO admin_role;
Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Utilisateur TO admin_role;
Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Fournisseur TO admin_role;
Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Stock TO admin_role;
Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Produit TO admin_role;
Grant succeeded.
```

```
SQL> GRANT CREATE SESSION TO livreur_role;
Grant succeeded.

SQL> GRANT SELECT ON Livraison TO livreur_role;
Grant succeeded.

SQL> GRANT UPDATE ON Livraison TO livreur_role;
Grant succeeded.
```

The image features decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines in a light teal color, with a curved line segment extending from them. The bottom-right corner features a large, light teal circular arc and several parallel diagonal lines.

06 - TESTER LES PRIVILEGES.

```
SQL> ^C
C:\Users\Dark>sqlplus livreur_user/livreur_password@localhost/xe

SQL*Plus: Release 11.2.0.2.0 Production on Sat Jun 8 16:37:12 2024

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> SELECT * FROM mahalcomtest.Livraison;

ID_LIVRAISON DATE_LIV STATUT
-----
ID_COMMANDE ID_LIVREUR
-----
            1 24-06-09 En cours
            1          1

SQL> SELECT * FROM mahalcomtest.Utilisateur;
SELECT * FROM mahalcomtest.Utilisateur
                                *
ERROR at line 1:
ORA-00942: table or view does not exist
```





PROBLEMS **ET** SOLUTIONS

PROBLEME



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dark>sqlplus mahalcomtest/0000@localhost/xe

SQL*Plus: Release 11.2.0.2.0 Production on Sat Jun 8 15:42:25 2024

Copyright (c) 1982, 2014, Oracle. All rights reserved.


Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE ROLE admin_role;
CREATE ROLE admin_role
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

SOLUTION



```
C:\Users\Dark>sqlplus system/0000@localhost/xe
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Sat Jun 8 15:50:43 2024
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

```
SQL> GRANT CREATE USER TO mahalcomtest;
```

```
Grant succeeded.
```

```
SQL> GRANT CREATE ROLE TO mahalcomtest;
```

```
Grant succeeded.
```

```
SQL> GRANT GRANT ANY PRIVILEGE TO mahalcomtest;
```

```
Grant succeeded.
```

The image features a central text element surrounded by four decorative geometric patterns in the corners. The top-left corner contains a series of parallel diagonal lines in a light blue-grey color. The top-right corner features a cluster of overlapping semi-circles in yellow, dark blue, red, and teal. The bottom-left corner also features a cluster of overlapping semi-circles in red, teal, dark blue, and red. The bottom-right corner contains a series of parallel diagonal lines in a light blue-grey color, mirroring the top-left pattern.

CONCLUSION



Dans ce processus, nous avons réussi à configurer des rôles, des utilisateurs et des privilèges dans une base de données Oracle.

Cette approche structurée simplifie la gestion des utilisateurs et renforce la sécurité de la base de données en contrôlant l'accès au niveau des rôles.

The image features a central text element surrounded by four decorative geometric patterns in the corners. The top-left pattern consists of a series of parallel diagonal lines. The top-right pattern is composed of several overlapping semi-circles in shades of orange, teal, and dark blue. The bottom-left pattern also features overlapping semi-circles in red, teal, and dark blue. The bottom-right pattern includes a large semi-circle outline and a cluster of parallel diagonal lines.

MERCI