

MINI PROJET

# Administration de base de données

Enseignante : Mme Leila Bayoudhi

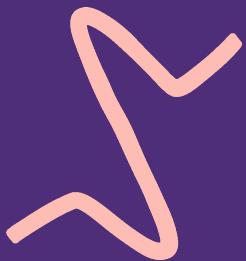
# Rencontrez notre Équipe



**Hadil masmoudi**



**Malek bentati**



**Groupe: LSL2 1TP1**

# Sommaire

- \* PARTIE 1: cahier de charge
- \* PARTIE 2: Modèle entité/association
- \* PARTIE 3: Diagramme de classe
- \* PARTIE 4: Implementation

# GESTION D'UNE LIBRARIE

# Exemple de cahier des charges

Copyright www.mindyour-go.com

# CAHIER DE CHARGE

Contenu et définition du problème

1

2

Objectif du projet

Nom du projet	librairie des collègues
Responsables	Hadil Masmoudi & Malek Bentati
Contexte et définition du problème	la Librairie des collègues est une entreprise familiale qui commercialise des outils scolaires. Dans un contexte de plus en plus concurrentiel , l' entreprise souhaite se développer .Déjà très occupé par ses activités de production , le chef d' entreprise est conscient de délaisser l' aspect commercial. Il envisage alors de confier cet aspect à des étudiants.
Objectif du projet	Augmenter le chiffre d'affaire de la librairie et donner l'opportunité de travailler à temps partiel pour les étudiants.
Echéance	Le 3 MAI 2022

## périmètre du projet

le chef du projet interagit avec :

- les fournisseurs: pour commander les produits
- les employés : pour contrôler les produits
- les livreurs : pour livrer les produits
- les clients: pour acheter les produits

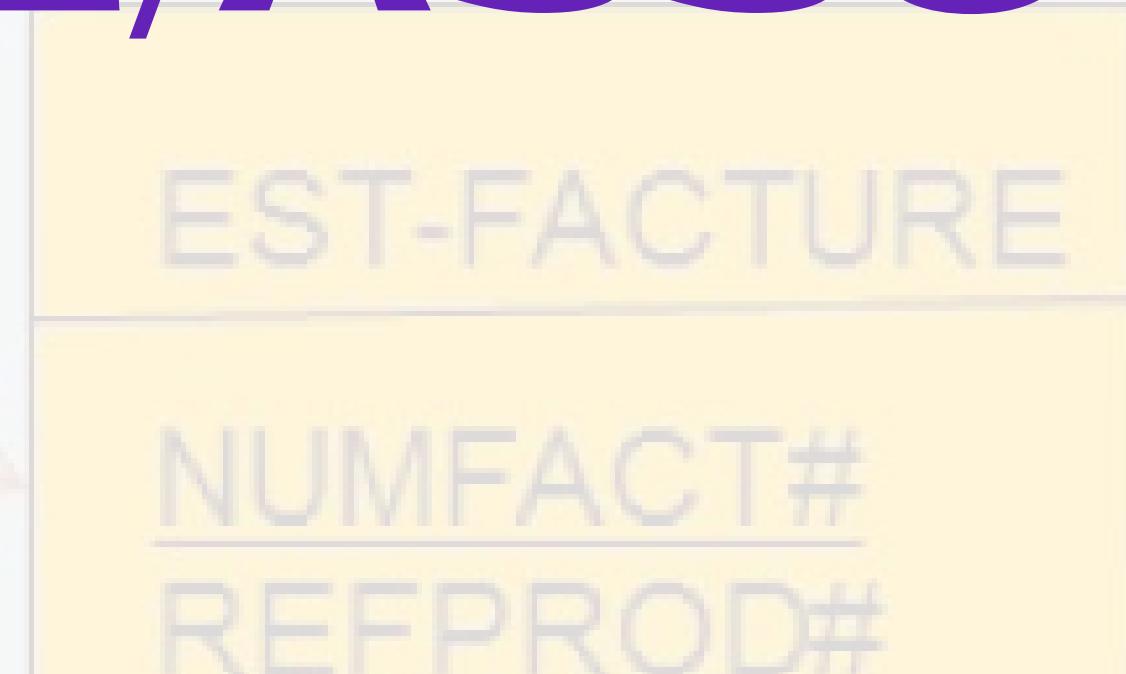
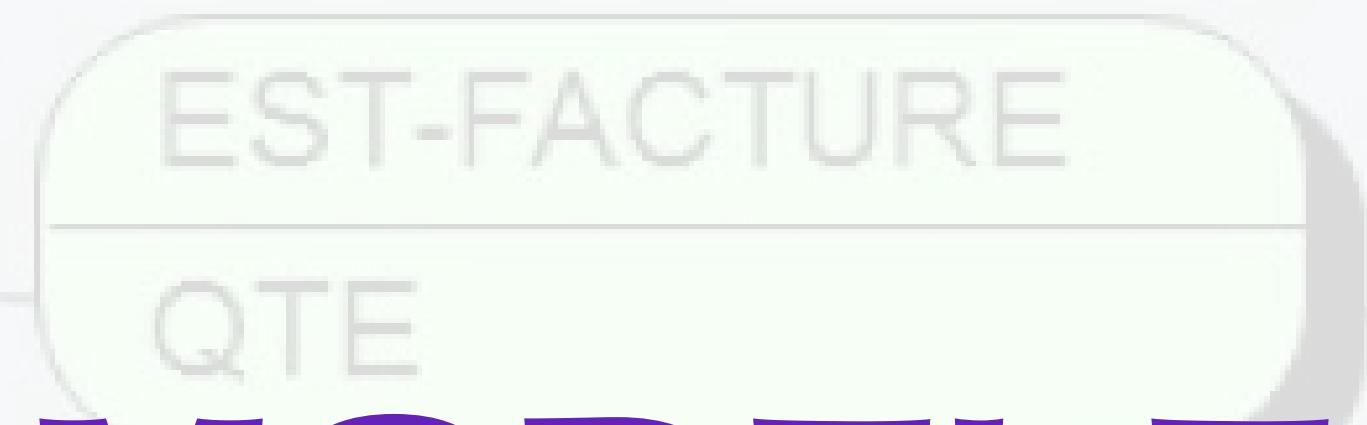
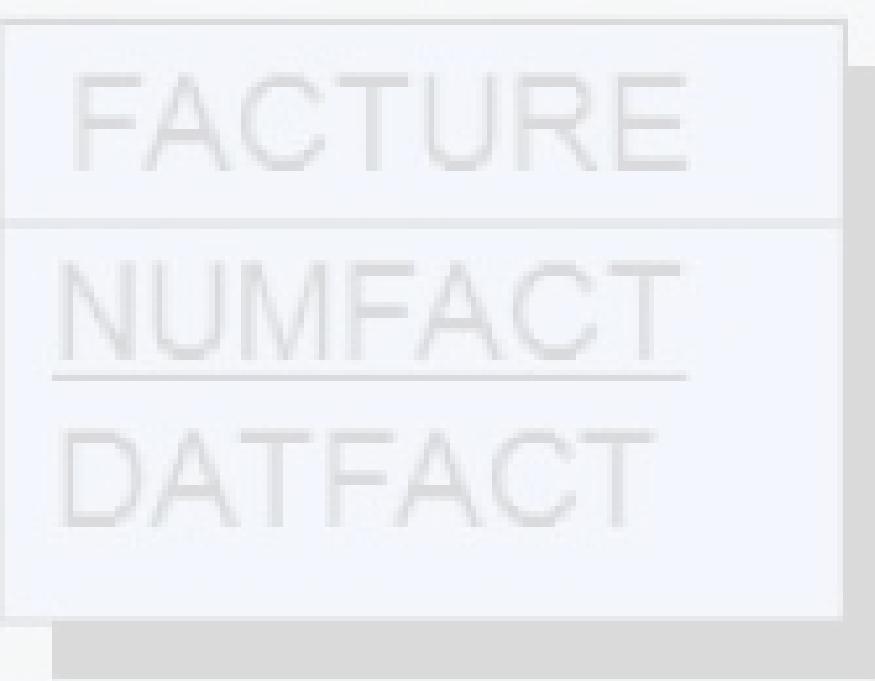
## périmètre hors projet

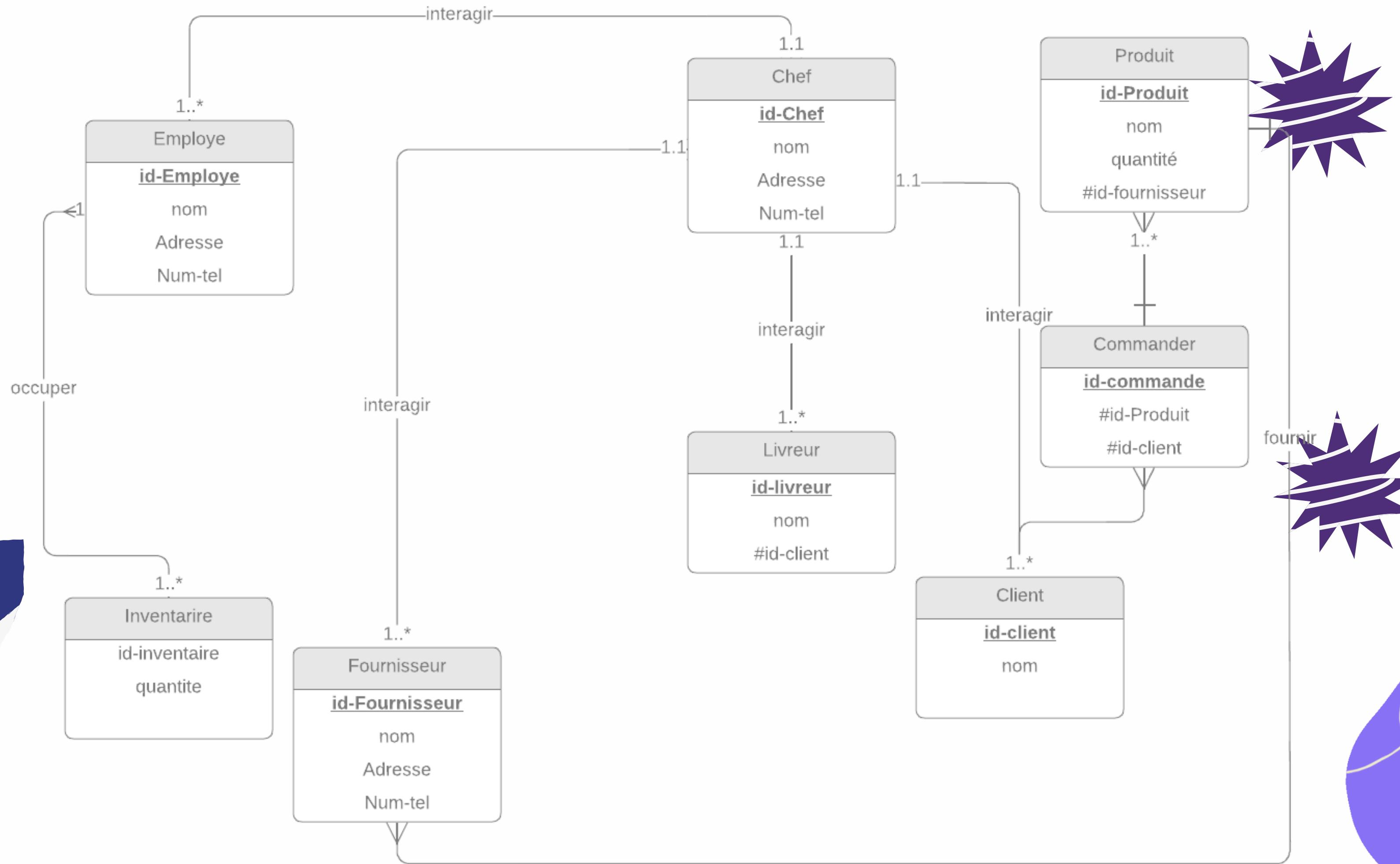
l inventaire s' occupe par les employés

## Enveloppe Budgetaire

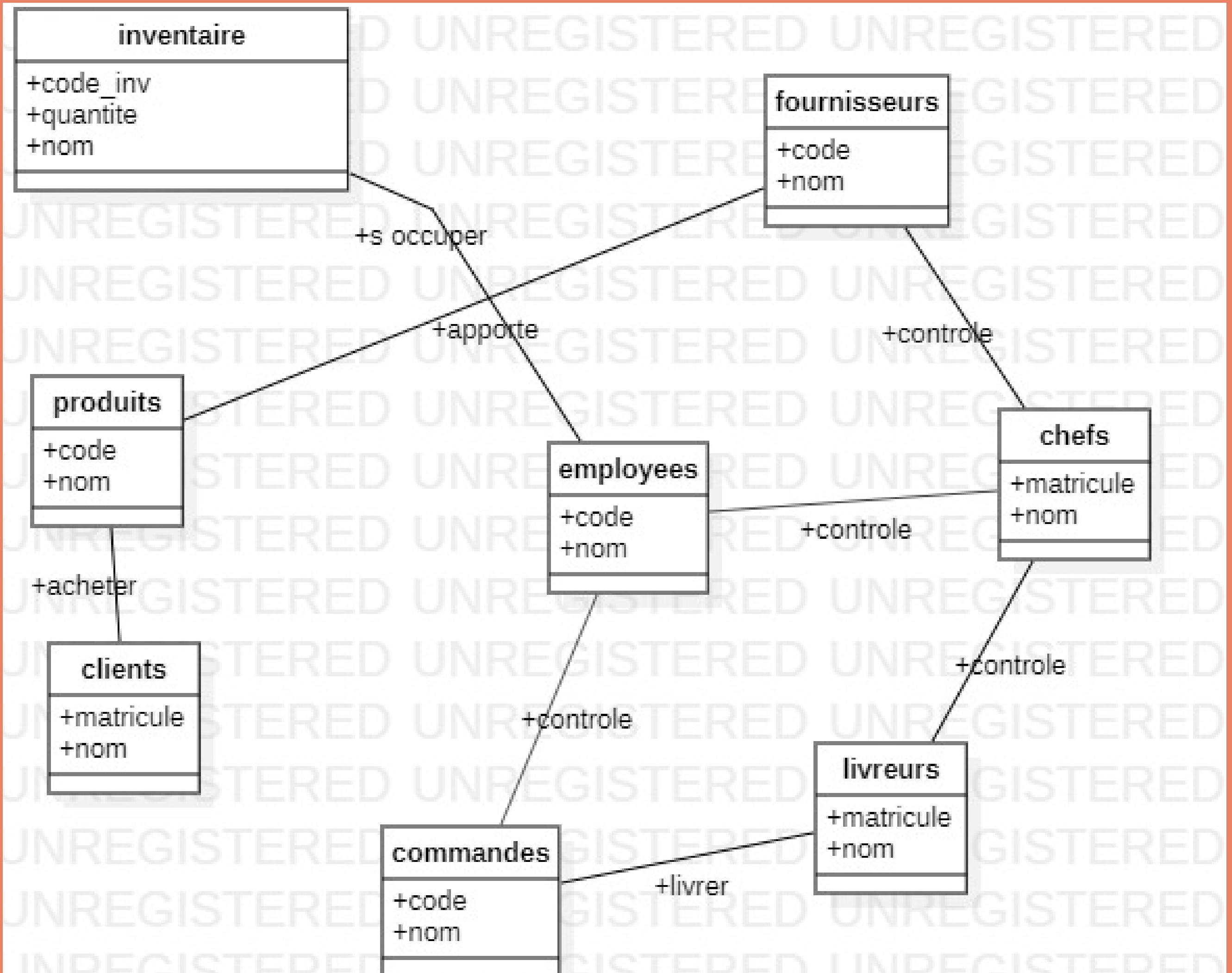
- Le budget total pour le projet ne doit pas dépasser 10 Million Dinars.

# **MODELE RELATIONNELLE ENTITE/ASSOCIATION**

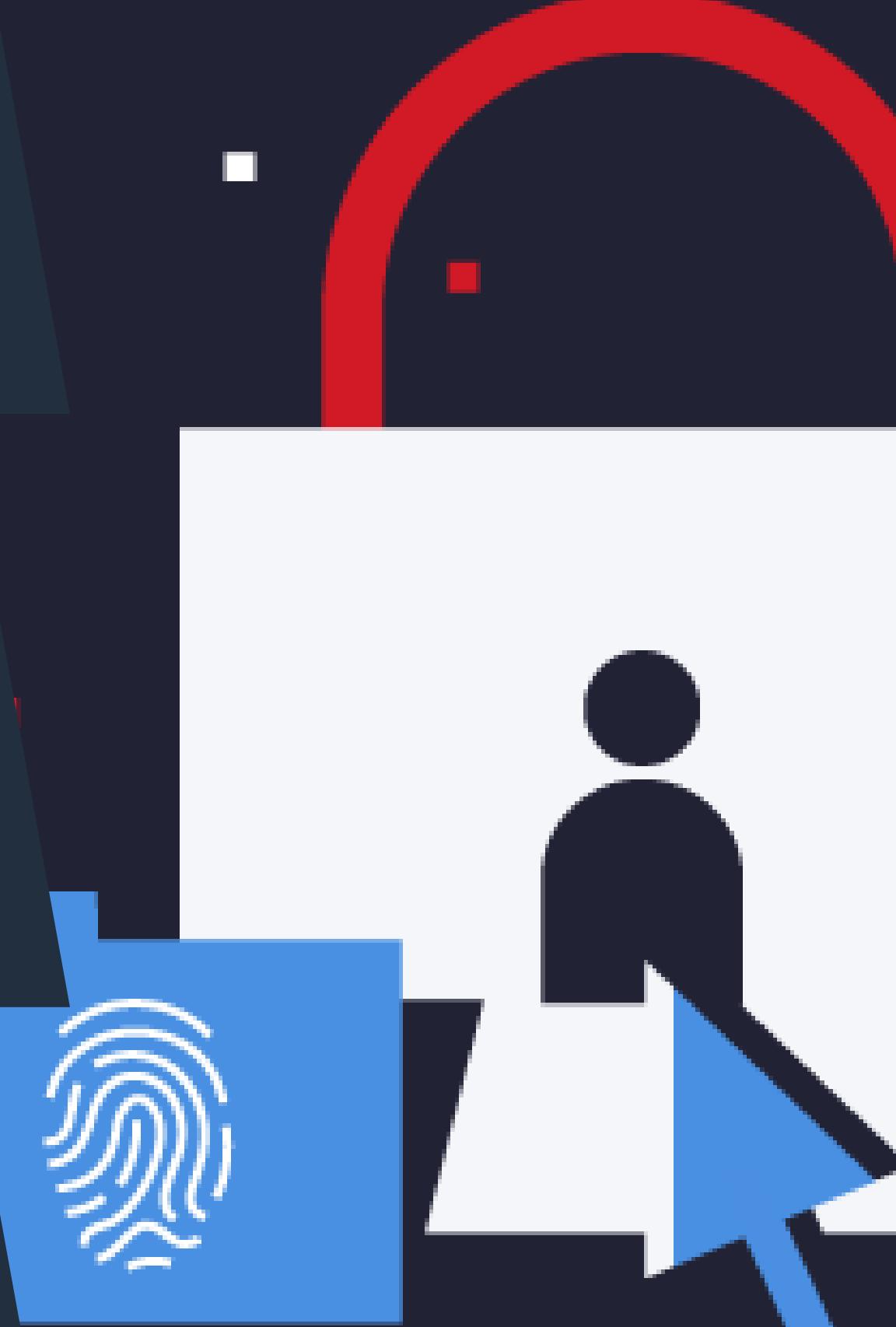




# DIAGRAMME DE CLASSE



# Gestion des profiles et des utilisateurs



## Lancer sqlplus et connecter en tant que administrateur

```
C:\app\LENOVO\product\21c\dbhomeXE\bin\sqlplus.exe

SQL*Plus: Release 21.0.0.0.0 - Production on Lun. Mai 2 17:47:16 2022
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Entrez le nom utilisateur :

Entrez le nom utilisateur : /as sysdba

Connecté :
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
```

```
SQL> alter session set "_oracle_script"=true;

Session modifiée.
```

Création du profile app\_chef ayant les limitations suivantes avec la vérification:

- Après cinq tentatives échouées de connexions, le compte sera verrouillé.
  - Le mot de passe sera expire après 30 jours.
  - Le même mot de passe ne sera pas ré-utilisé pendant trois minutes.

Création du profile app\_employee ayant les limitations suivantes:

- Après cinq tentatives échouées de connexions, le compte sera verrouillé.
- Le nombre maximale de blocs lus durant une session 100000.
- Le nombre maximale de blocs lus durant un appel serveur 100.
- La durée de connexion maximale d'une session 15 minutes .
- Temps d'inactivité(en minutes) autorisé 5 minutes.
- Le temps de processeur maximum (en centièmes de secondes )
- Qu'une session puisse utiliser :2 minutes .

```
SQL> create profile app_employee limit  
  2  failed_login_attempts 2  
  3  logical_reads_per_session 100000  
  4  logical_reads_per_call 100  
  5  connect_time 15  
  6  idle_time 5  
  7  cpu_per_session 2;
```

Profil crée.

verification:

```
SQL> select profile from dba_profiles where profile='APP_EMPLOYEE';
```

PROFILE

```
-----  
APP_EMPLOYEE  
APP_EMPLOYEE
```

18 lignes sélectionnées.

## creation des utilisateurs <chef,employee,fournisseur,client,livreur>

```
SQL> create user chef identified by chef;  
Utilisateur crée.  
  
SQL> create user employee identified by employee;  
Utilisateur crée.  
  
SQL> create user fournisseur identified by fournisseur;  
Utilisateur crée.  
  
SQL> create user client identified by client;  
Utilisateur crée.  
  
SQL> create user livreur identified by livreur;  
Utilisateur crée.
```

Affectez l'utilisateur chef et employee aux profils “app\_chef” et “app\_employee” .

```
SQL> alter user chef profile app_chef;  
Utilisateur modifié.  
  
SQL> alter user employee profile app_employee;  
Utilisateur modifié.
```

Connecter à user chef tout en fournissant un mot de passe incorrect.on va Essayer ce traitement cinq fois de suite.Puis on va reconnecter en fournissant le mot de passe correct.

```
SQL> conn chef/cef
ERROR:
ORA-01017: nom utilisateur/mot de passe non valide ; connexion refusée

Avertissement : vous n'êtes plus connecté à ORACLE.
SQL> conn chef/ccc
ERROR:
ORA-01017: nom utilisateur/mot de passe non valide ; connexion refusée

SQL> conn chef/cheef
ERROR:
ORA-01017: nom utilisateur/mot de passe non valide ; connexion refusée

SQL> conn chef/1
ERROR:
ORA-01017: nom utilisateur/mot de passe non valide ; connexion refusée

SQL> conn chef/cheff
ERROR:
ORA-01017: nom utilisateur/mot de passe non valide ; connexion refusée

SQL> conn chef/chef
ERROR:
ORA-28000: Compte verrouillé.
```

vérification du compte qui est verrouillée afin de dé résolu le problème.

```
SQL> conn /as sysdba  
Connecté.
```

```
SQL> select account_status from dba_users where username='CHEF';  
  
ACCOUNT_STATUS  
-----  
LOCKED(TIMED)
```

```
SQL> alter user chef account unlock;  
Utilisateur modifié.  
  
SQL> select account_status from dba_users where username='CHEF';  
  
ACCOUNT_STATUS  
-----  
OPEN
```

# Gestion des roles



Création d'un rôle appelé on\_session, qui permet de :

- de créer une session

```
SQL> create role on_session;  
Rôle crée.  
  
SQL> grant create session to on_session;  
Autorisation de priviléges (GRANT) acceptée.
```

connecter en tant que employee et résoudre le problème :

```
SQL> conn employee/employee  
ERROR:  
ORA-01045: l'utilisateur EMPLOYEE n'a pas le privilège CREATE SESSION ;  
connexion refusée
```

```
SQL> show user  
USER est ""  
SQL> conn /as sysdba  
Connecté.  
SQL> grant on_session to employee;  
  
Autorisation de priviléges (GRANT) acceptée.
```

Création d'un rôle appelé admin, qui permet de :

- de se connecter,
- de créer une table

```
SQL> create role admin ;
```

Rôle crée.

```
SQL> grant connect to admin;
```

Autorisation de priviléges (GRANT) acceptée.

```
SQL> grant create table to admin ;
```

Autorisation de priviléges (GRANT) acceptée.

Affectez le rôle ADMIN à l'utilisateur "chef".

```
SQL> grant admin to chef;
```

Autorisation de priviléges (GRANT) acceptée.

Création du rôle allabouttablespaces , qui permet de:

- de créer , modifier et supprimer un tablespace.

```
SQL> create role allabouttablespaces;
```

Rôle crée.

```
SQL> grant DROP TABLESPACE,ALTER TABLESPACE,CREATE TABLESPACE to allabouttablespaces;
```

Autorisation de priviléges (GRANT) acceptée.

Create role aboutuser, qui permet de:

- de créer , modifier et supprimer un utilisateur.

```
SQL> create role aboutuser;
```

Rôle créé.

```
SQL> grant create user,drop user,alter user to aboutuser;
```

Autorisation de priviléges (GRANT) acceptée.

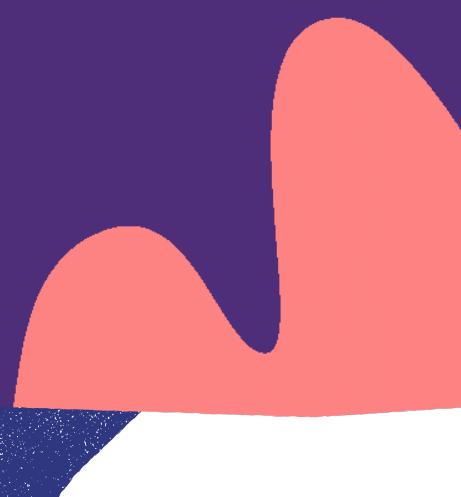
Affecter les rôles allabouttablespaces et aboutuser au rôle "admin".

```
SQL> grant allabouttablespaces to admin;
```

Autorisation de priviléges (GRANT) acceptée.

```
SQL> grant aboutuser to admin;
```

Autorisation de priviléges (GRANT) acceptée.



connecter en tant que chef et supprimer l'utilisateur client .

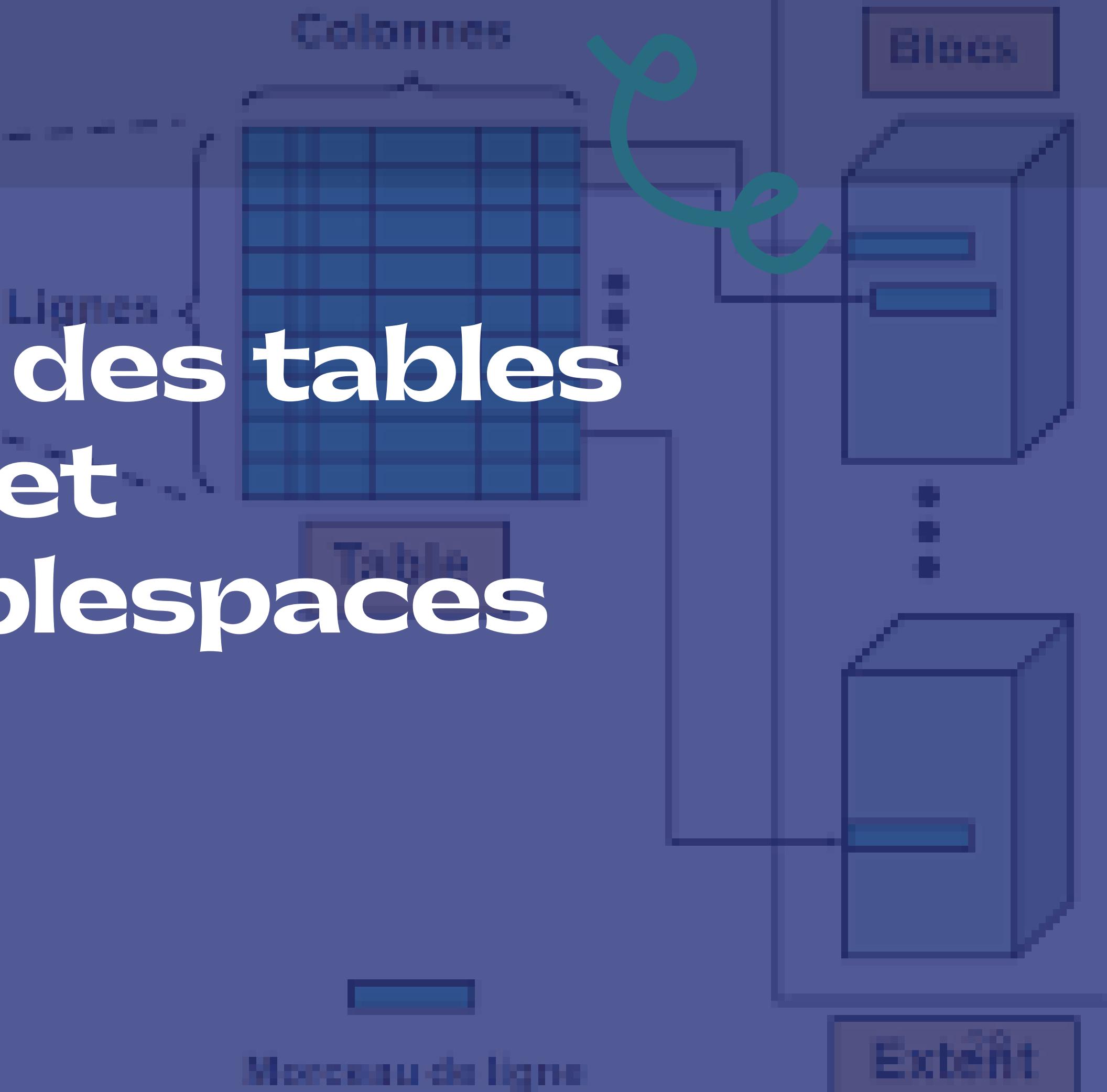
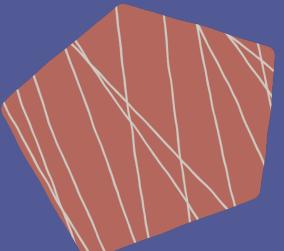
```
SQL> conn chef/chef;  
Connecté.
```

```
SQL> drop user client cascade;  
Utilisateur supprimé.
```

verifier la suppression.

```
SQL> conn /as sysdba  
Connecté.  
  
SQL> select username from all_users;  
  
USERNAME  
-----  
C##ETUDIANT  
U1  
U2  
CHEF  
EMPLOYEE  
FOURNISSEUR  
LIVREUR  
  
51 lignes sélectionnées.
```

# Gestion des tables et des tablespaces



Moniteur de ligne

Exécution

1/creer un table f et inserer des lignes.

```
SQL> create table f(code number,nom varchar(20));  
Table crée.  
  
SQL> grant insert on chef.f to admin;  
Autorisation de priviléges (GRANT) acceptée.  
  
SQL> alter session set "_ORACLE_SCRIPT"=true;  
Session modifiée.  
  
SQL> ALTER USER chef quota unlimited on users;  
Utilisateur modifié.
```

3/verifier la selection:

```
SQL> conn employee/employee  
Connecté.  
SQL> select * from chef.f  
2 /  
  
CODE NOM  
-----  
1 bic  
2 schneider  
3 parker  
4 stablio boss  
5 casio  
6 pensan  
7 reynold  
  
7 lignes sélectionnées.
```

```
SQL> insert into f(code,nom) values (1,'bic');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (2,'schneider');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (3,'parker');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (4,'stablio boss');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (5,'casio');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (6,'pensan');  
1 ligne créée.  
  
SQL> insert into f(code,nom) values (7,'reynold');  
1 ligne créée.
```

2/chef donne à employee le droit d'interroger (droit select) sa table "f".

```
SQL> grant select on f to employee;  
Autorisation de priviléges (GRANT) acceptée.
```

creation de tablespace:

- Nom : produit
- Nombre de fichiers : 1
- Noms de fichiers : « prod.dbf »
- La taille initiale du fichier : 256K
- Fichier extensible avec : 128K
- La taille maximale du fichier : 2M
- Gestion d'extents : Local AUTOALLOCATE
- L'emplacement du fichier : 'C:\projet base'

```
SQL> conn chef/chef
Connecté.
SQL> create tablespace produit datafile 'C:\projet base\prod.dbf' size 256K autoextend on next 128K maxsize 2M
  2 extent management local autoallocate;
Tablespace créé.
```

creation d'une nouvelle table tab\_prod dans le tablespace « produit » et inserer des données:

```
SQL> create table tab_prod(code number,nom varchar(20)) tablespace produit;
Table créé.
```

```
SQL> ALTER USER chef quota unlimited on produit;
Utilisateur modifié.
```

```
SQL> insert into tab_prod(code,nom) values (10,'stylo');

1 ligne créée.
```

```
SQL> insert into tab_prod(code,nom) values (20,'marqueur');

1 ligne créée.
```

```
SQL> insert into tab_prod(code,nom) values (30,'cahier');

1 ligne créée.
```

```
SQL> insert into tab_prod(code,nom) values (40,'calculatrice');

1 ligne créée.
```

# Gestion des privileges



chef donne à employeur le droit d'insérer des données dans son table tab\_prod.

```
SQL> ALTER USER chef quota unlimited on produit;
```

```
Utilisateur modifi .
```

```
SQL> grant insert on tab_prod to employee;
```

```
Autorisation de privil ges (GRANT) accept e.
```

```
SQL> conn employee/employee  
Connect .
```

```
SQL> insert into chef.tab_prod(code,nom) values(50,'gomme');
```

```
1 ligne cr  e.
```

```
SQL> conn employee/employee  
Connect .
```

```
SQL> insert into chef.tab_prod(code,nom) values(50,'gomme');
```

```
1 ligne cr  e.
```

chef donne à employeur le droit d'interroger son table tab\_prod.

```
SQL> grant select on tab_prod to employee;
```

```
Autorisation de priviléges (GRANT) acceptée.
```

```
SQL> conn employee/employee  
Connecté.
```

```
SQL> select * from chef.tab_prod;
```

CODE	NOM
10	stylo
20	marqueur
30	cahier
40	calculatrice
50	gomme

Merci !