

# Rapport de Projet : Administration Base de Données

---

# ORACLE

01/11/2023

---

Réalise par : Sofiane ABETIOU & Hajar KDIDAR  
Encadre : Hassan BADIR

ORACLE®

---

# INTRODUCTION

La gestion efficace des achats et des employés est un élément essentiel de la réussite d'un centre commercial prospère. Dans un environnement dynamique où les consommateurs recherchent constamment des expériences de shopping exceptionnelles, la gestion de ces deux aspects revêt une importance capitale. Le présent cahier des charges se penche sur la conception et le développement d'un système de gestion des achats et des employés spécifiquement conçu pour répondre aux besoins

## Base de Données

Au cœur de ce projet se trouve une base de données robuste et performante qui permettra de stocker, gérer et analyser de manière sécurisée les informations liées aux achats, aux stocks, aux employés, et aux clients. Une base de données bien conçue offre une visibilité et un contrôle accrus sur l'ensemble des opérations du centre commercial.

## Sécurité de la Base de Données

La sécurité des données est une préoccupation majeure dans notre ère numérique. La base de données contiendra des informations sensibles, notamment des données financières et des informations personnelles sur les employés et les clients. Il est impératif de mettre en place des mesures de sécurité rigoureuses pour protéger ces données contre les menaces telles que les accès non autorisés, les fuites d'informations, et les perturbations potentielles.

## L'Importance de la Sécurité des Données

La sécurité des données est un pilier fondamental de tout projet informatique. Elle garantit la confidentialité, l'intégrité et la disponibilité des informations, tout en renforçant la confiance des clients et la

conformité aux réglementations en vigueur. Dans le contexte de la gestion des achats et des employés d'un centre commercial, la sécurité des données revêt une importance capitale pour protéger les intérêts de toutes les parties prenantes et maintenir la réputation du centre commercial.

## Technologie

**Notre décision de retenir Oracle comme système de gestion de base de données repose sur sa réputation inégalée en matière de performance, de fiabilité et de sécurité. L'architecture bien structurée d'Oracle garantit une gestion efficace des données, en offrant une flexibilité exceptionnelle pour s'adapter à nos besoins évolutifs. La robustesse de son moteur de base de données assure une intégrité des données inébranlable, tandis que ses fonctionnalités avancées de sécurité et de gestion des transactions renforcent la confiance dans la gestion des privilèges. Oracle est le choix privilégié pour soutenir notre infrastructure, offrant une base solide pour nos opérations informatiques tout en garantissant une gestion transparente de nos données.**



## Chapitre 2 : Conception

### 1- Cahier de charge

Pour spécifier les besoins et les fonctionnalités de notre projet, nous avons préparé un cahier des charges détaillé qui servira de document central pour aligner toutes les parties prenantes sur les

---

objectifs et les attentes du projet. Ce cahier de charges, soigneusement élaboré, décompose le projet en éléments clairement définis, en mettant en lumière les besoins essentiels du centre commercial, les fonctionnalités clés du système à développer, les exigences techniques, et les priorités.

En créant ce cahier de charges, notre objectif est de fournir un cadre solide pour le processus de développement, en garantissant une compréhension commune des besoins du projet parmi l'équipe de développement, les clients, les utilisateurs finaux et toutes les parties impliquées.

Ce document détaillé favorise la transparence, la cohérence et la gestion efficace des attentes, ce qui est essentiel pour assurer la réussite du projet de gestion des achats et des employés du centre commercial. En outre, il constitue un point de départ essentiel pour l'évaluation de la conformité du projet aux besoins et aux objectifs initialement définis, contribuant ainsi à la satisfaction des parties prenantes et à la livraison d'une solution robuste et efficace.

**Le cahier des charges complet se trouve dans le dossier du projet**

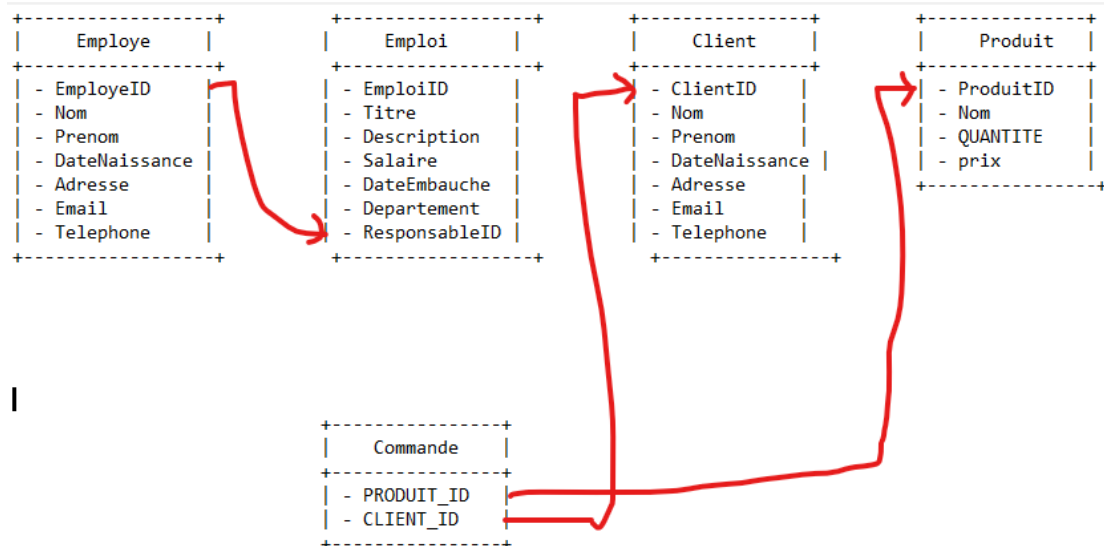
## **2- Modèle de Données Logiques (MLD)**

Un Modèle de Données Logiques (MLD) est un outil de modélisation qui permet de représenter la structure interne d'un système en se concentrant sur les entités (ou tables) qui composent la base de données. Dans un MLD, chaque entité est associée à des attributs (champs de la table) qui décrivent les propriétés de ces entités. De plus, les relations entre les entités, généralement exprimées par les liaisons de clés étrangères, indiquent comment ces entités sont liées les unes aux autres.

---

Dans le contexte de la gestion des achats et des employés d'un centre commercial, un MLD permettrait de représenter les tables pertinentes, telles que les tables Employes, Emplois, Clients, Produit et Commande, et d'indiquer les attributs associés à chaque table. Par exemple, la table Employes comprendrait des attributs tels que EmployeID, Nom, Prenom, DateNaissance, etc. De plus, les relations entre ces tables, telles que la clé étrangère dans la table Emplois faisant référence à la table Employes, seraient spécifiées.

Le MLD permet de créer une vue structurée des données et des relations entre elles, ce qui facilite la conception et la gestion d'une base de données. Cela permet également aux équipes de développement et aux professionnels de la base de données de comprendre comment les données sont organisées et comment elles interagissent au sein du système, ce qui est essentiel pour la planification, la communication, et la conception d'une application ou d'un système de gestion de données.



### 3-Plan de travail :

#### 1- Création d'utilisateurs

#### 2- Création de rôles et privilèges par département

##### 2.1 – Département administration

###### 2.1.1- Admins

###### 2.1.2 – Secrétaire

##### 2.2 – Département RH

###### 2.2.1 – Responsable RH

###### 2.2.2 - Vice\_responsable RH

##### 2.3 – Département Achat

###### 2.3.1 - Responsable d'achat

###### 2.3.2 – Caissier

##### 2.4 - Département Client

#### 3 – Création des profiles

Les profils Oracle sont utilisés pour définir des paramètres de ressources spécifiques pour les utilisateurs ou les rôles, ce qui peut contrôler leur utilisation des ressources

---

du système de gestion de base de données Oracle. Voici une brève explication de chaque profil que vous avez créé :

**PROFIL\_ADMIN**

**PROFIL\_SECR**

**PROFIL\_RH**

**PROFIL\_RACHAT**

**PROFIL\_CAISSIER**

**PROFIL\_CLIENT**

## ⇒ Explication de conception

### **Création des Utilisateurs et Attribution des Privilèges**

**Création des Utilisateurs** : La première étape consiste à créer des utilisateurs dans la base de données Oracle. Chaque utilisateur est associé à un département spécifique de l'organisation, tel que l'administration, les ressources humaines, les achats, etc. Chacun d'entre eux se voit attribuer un nom d'utilisateur, un mot de passe, un espace de stockage par défaut et d'autres paramètres de configuration.

**Attribution des Privilèges** : Ensuite, des privilèges sont attribués à chaque utilisateur en fonction de son rôle et de son département. Par exemple, l'administrateur, représenté par l'utilisateur ADMINST1, reçoit des privilèges étendus, tels que la création de tables et de vues, tandis que d'autres utilisateurs reçoivent des privilèges plus limités, comme **CONNECT** ou **RESOURCE**.



---

## Création des Rôles et Attribution des Privilèges par Département

**Département Administration :** Pour le département administratif, des rôles spécifiques sont créés, tels que RSECR pour les secrétaires. Chaque rôle est attribué à des utilisateurs ayant des fonctions similaires. Par exemple, Le rôle RSECR est attribué à l'utilisateur SECR, ce qui détermine les privilèges dont il dispose, notamment le droit de voir le résultat d'une vue appelée **Vue\_EmployeEmploi**.

**Département RH :** Dans le département des ressources humaines, des rôles sont créés pour les responsables RH (RRH) auxquels sont accordés les droits de sélection (SELECT), de mise à jour (UPDATE), et d'insertion (INSERT) sur les tables **Employes** et **Emplois**. De plus, les responsables RH ont le privilège de donner des droits au rôle RVRH, leur permettant uniquement de voir le résultat de la vue **Vue\_EmployeEmploi**. Les **Vue\_EmployeEmploi**). Ces rôles sont attribués aux utilisateurs respectifs, comme le rôle RRH attribué à l'utilisateur RESPO\_RH.

**Département Achat :** Le département des achats peut avoir des rôles spécifiques attribués aux membres de l'équipe. Le rôle du responsable des achats (RACHAT) inclut les droits de sélection (SELECT), de mise à jour (UPDATE), de suppression (DELETE), et d'insertion (INSERT) sur les tables Clients, Produits, et Commande. Les caissiers (Rcaissier) disposent uniquement du droit de voir les données des clients (SELECT FROM Clients), d'insérer des informations dans la table Commande (INSERT INTO Commande), et de sélectionner des données à partir de la table Commande (SELECT FROM Commande). Ces rôles sont attribués aux utilisateurs appropriés, tels que l'utilisateur UACHAT.

Quant aux clients, ils ont uniquement le droit de sélection (SELECT) sur la table Produits et la possibilité de voir le résultat de la vue

**VueCommandes**, ce qui signifie qu'ils peuvent consulter les produits disponibles et les commandes passées sans la possibilité de les modifier.



---

## Création des Profils et Attribution aux Utilisateurs

**Création des Profils :** Des profils sont créés avec des limites spécifiques pour contrôler l'utilisation des ressources par les utilisateurs. Par exemple, le profil PROFIL\_ADMIN est illimité, tandis que d'autres profils, tels que PROFIL\_SECR, PROFIL\_RH, PROFIL\_RACHAT, PROFIL\_CAISSIER, et PROFIL\_CLIENT, imposent des limites sur des ressources telles que le nombre de sessions, le temps CPU, le temps de connexion, etc.

**Attribution des Profils aux Utilisateurs :** Enfin, les profils sont attribués aux utilisateurs correspondants. Par exemple, l'administrateur se voit attribuer le profil PROFIL\_ADMIN, garantissant un accès complet sans restrictions. Les responsables RH et d'autres utilisateurs reçoivent des profils spécifiques, adaptés à leurs besoins.

### Importance de la Sécurité

Cette structure de gestion des utilisateurs, des rôles, des privilèges et des profils est essentielle pour garantir la sécurité de la base de données. Elle permet de contrôler l'accès aux données, de protéger les informations sensibles et de prévenir les abus. De plus, elle garantit une utilisation équilibrée des ressources de la base de données, contribuant ainsi à une performance stable. La sécurité est un aspect clé de la gestion des bases de données, permettant de répondre aux normes de conformité et de garantir la protection des données de l'entreprise contre tout accès non autorisé.

---

## Chapitre 3 ----- IMPLEMENTATION && TEST

### Introduction :

Lors de l'implémentation, notre engagement premier est de demeurer en étroite conformité avec le cahier des charges établi. En suivant scrupuleusement les spécifications définies, notre objectif est de garantir la satisfaction du client. Cette démarche nous permet de répondre au maximum des attentes et d'assurer que notre solution soit parfaitement adaptée à ses besoins.

### Implémentation :

La rédaction de ce chapitre suit une méthodologie particulière « ***tester chaque implémentation de privilège*** » ».

L'objectif est clair : garantir la sécurité et la fiabilité. Cette approche proactive vise à éliminer **les failles de sécurité** et à assurer **une gestion des privilèges efficace**.

Au fil des pages, nous détaillerons notre méthodologie de test pour chaque privilège, mettant en lumière nos efforts pour renforcer la sécurité de notre système.

*\*NB : Le lien du code source vous le trouver avec les documents envoyer \*:*

L'implémentation suit rigoureusement le plan de travail détaillé précédemment exposé dans la première section du chapitre 1.

### **1-creation des utilisateurs :**

User administrateur :

```
SQL Plus
SQL> show user
USER est "SYS"
SQL>
SQL> CREATE USER ADMINST1 IDENTIFIED BY ADMINST1 DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.

SQL>
```

#### User secrétaire :

```
SQL> CREATE USER SECR IDENTIFIED BY SECR DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.

SQL>
```

#### département Ressource humaine :

##### User responsable :

```
SQL> CREATE USER RESPO_RH IDENTIFIED BY RERH DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.

SQL>
```

##### User vice responsable :

```
SQL> CREATE USER VICE_RESPO_RH IDENTIFIED BY VCRERH
2 DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA
3 UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.

SQL>
```

#### département Achat :

##### User responsable && User caissier :

```
SQL> CREATE USER UACHAT IDENTIFIED BY UACHAT DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.

SQL> CREATE USER UCAISSER IDENTIFIED BY UCAISSER DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.
```

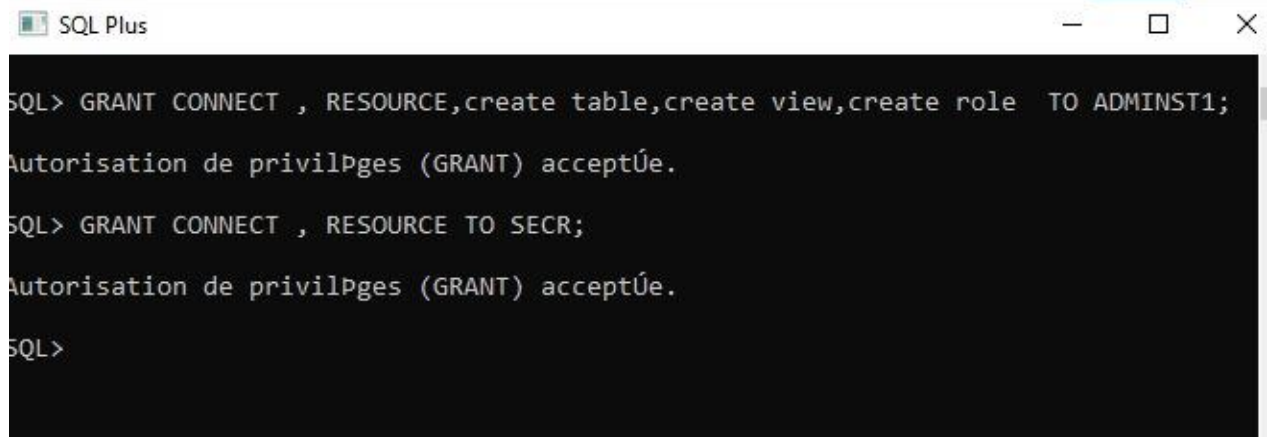
##### User client :

```
SQL> create user client identified by client DEFAULT TABLESPACE USERS TEMPORARY
TABLESPACE TEMP QUOTA UNLIMITED ON USERS ACCOUNT UNLOCK;

Utilisateur créé.
```

## 2-Accords des privilèges :

### Département Administration :



```
SQL> GRANT CONNECT , RESOURCE,create table,create view,create role TO ADMINST1;
Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT CONNECT , RESOURCE TO SECR;
Autorisation de privilèges (GRANT) acceptée.

SQL>
```

### Département Ressource Humaine

```
SQL> GRANT CONNECT , RESOURCE,create view TO RESPO_RH;
Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT CONNECT , RESOURCE TO VICE_RESPO_RH;
Autorisation de privilèges (GRANT) acceptée.
```

### Département Achat :

```
SQL> GRANT CONNECT, RESOURCE, create view TO UACHAT;
Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT CONNECT , RESOURCE TO UCAISSER;
Autorisation de privilèges (GRANT) acceptée.

SQL>
```

### Le client:

```
SQL> GRANT CONNECT, RESOURCE TO CLIENT;
Autorisation de privilèges (GRANT) acceptée.
```

## 2-Creation des tables :

-les tables sont créées dans le Schéma de l'administrateur

```

SQL> CREATE TABLE Employes (
2      EmployeID INT PRIMARY KEY,
3      Nom VARCHAR(50),
4      Prenom VARCHAR(50),
5      DateNaissance DATE,
6      Adresse VARCHAR(100),
7      Email VARCHAR(100),
8      Telephone VARCHAR(15)
9  );

Table cr   e.

SQL>
SQL> -- Table des emplois
SQL> CREATE TABLE Emplois (
2      EmploiID INT PRIMARY KEY,
3      Titre VARCHAR(50),
4      Description VARCHAR(50),
5      Salaire DECIMAL(10, 2),
6      DateEmbauche DATE,
7      Departement VARCHAR(50),
8      ResponsableID INT,
9      FOREIGN KEY (ResponsableID) REFERENCES Employes(EmployeID)
10 );

Table cr   e.

SQL> --depart.achat---
SQL> -- TABLE DES CLIENTS
SQL> CREATE TABLE CLIENTS (
2      clientID INT PRIMARY KEY,
3      Nom VARCHAR(50),
4      Prenom VARCHAR(50),
5      DateNaissance DATE,
6      Adresse VARCHAR(100),
7      Email VARCHAR(100),
8      Telephone VARCHAR(15)
);

SQL>
SQL> --produit
SQL> CREATE TABLE PRODUIT (
2      ProduitID INT PRIMARY KEY,
3      Nom VARCHAR(50),
4      QUANTITE INT,
5      prix NUMBER
6  );

Table cr   e.

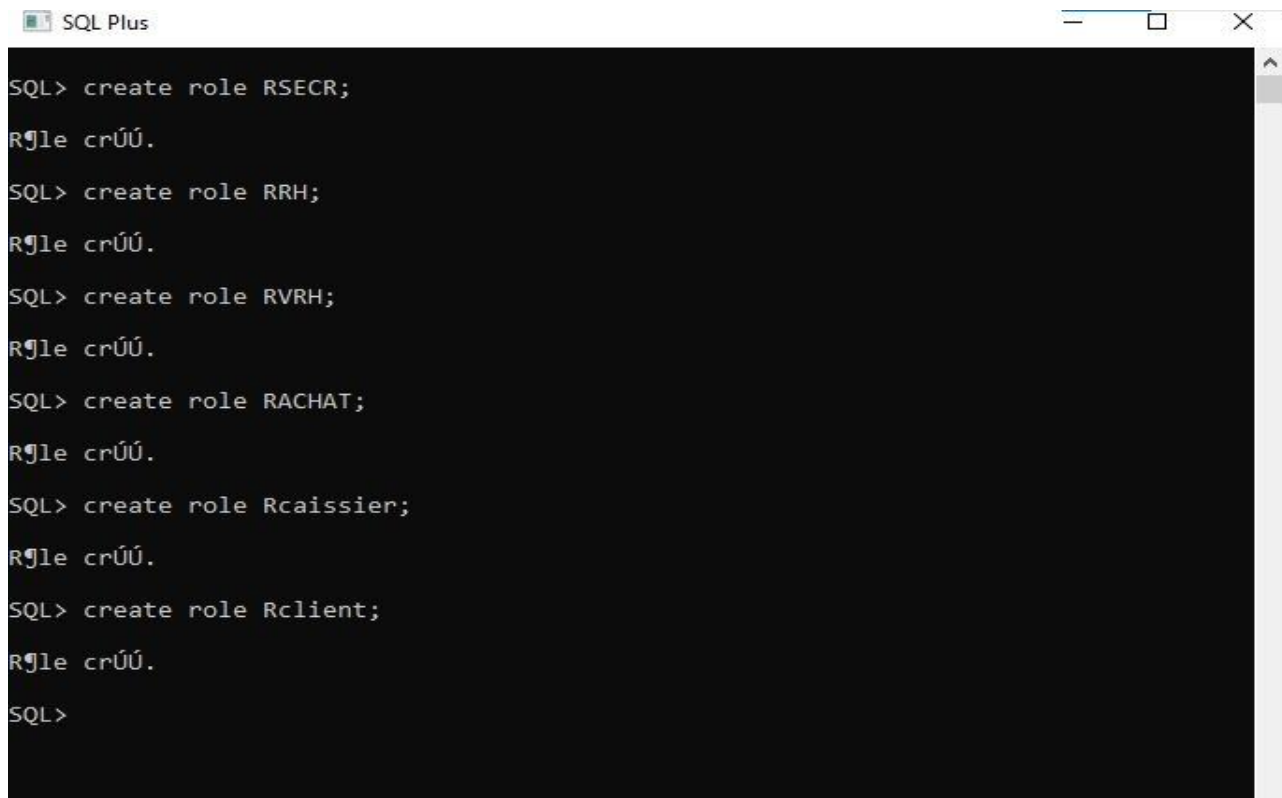
SQL>
SQL> --commande
SQL> CREATE TABLE COMMANDE (
2      PRODUIT_ID INT,
3      CLIENT_ID INT,
4      FOREIGN KEY (PRODUIT_ID) REFERENCES PRODUIT(ProduitID),
5      FOREIGN KEY (CLIENT_ID) REFERENCES CLIENTS(clientID)
6  );

Table cr   e.

```

## 2-creation des r  les

Tout en restant dans le sch  ma administrateur on cr  e les r  les .



```

SQL> create role RSECR;

R  le cr   e.

SQL> create role RRH;

R  le cr   e.

SQL> create role RVRH;

R  le cr   e.

SQL> create role RACHAT;

R  le cr   e.

SQL> create role Rcaissier;

R  le cr   e.

SQL> create role Rclient;

R  le cr   e.

SQL>

```

### 3-Peuplement des tables

```
SQL Plus
SQL> INSERT INTO ADMINST1.Employes (EmployeeID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES
  3 (1, 'Dupont', 'Jean', TO_DATE('1980-05-10', 'YYYY-MM-DD'), '123 Rue de la Paix', 'jean.dupont@email.com', '123-456-7890');
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Employes (EmployeeID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES (2, 'Martin', 'Sophie', TO_DATE('1990-08-15', 'YYYY-MM-DD'), '456 Avenue des Roses', 'sophie.martin@email.com', '987-654-3210');
1 ligne cr  e.

SQL> INSERT INTO ADMINST1.Employes (EmployeeID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES(3, 'Lefebvre', 'Paul', TO_DATE('1985-02-25', 'YYYY-MM-DD'), '789 Boulevard du Soleil', 'paul.lefebvre@email.com', '555-123-7890');
1 ligne cr  e.

SQL> INSERT INTO ADMINST1.Employes (EmployeeID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES(4, 'Dubois', 'Marie', TO_DATE('1995-11-03', 'YYYY-MM-DD'), '567 Chemin de la Lune', 'marie.dubois@email.com', '333-888-9999');
1 ligne cr  e.

SQL>
SQL>
SQL>
SQL> -- Insertion de donn  es dans la table ADMINST1.Emplois
SQL> -- Ins  rer des donn  es dans la table ADMINST1.Emplois
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (1, 'Ing  nieur logiciel', 'D  veloppement de logiciels', 80000.00, TO_DATE('2022-01-15', 'YYYY-MM-DD'), 'D  veloppement', 1);
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (2, 'Gestionnaire de projet', 'Gestion de projets informatiques', 90000.00, TO_DATE('2021-11-20', 'YYYY-MM-DD'), 'Gestion de projets', 2);
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (3, 'Sp  cialiste des ressources humaines', 'Gestion des RH', 60000.00, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 'Ressources humaines', 3);
1 ligne cr  e.
```

```
SQL>
SQL>
SQL>
SQL> -- Insertion de donn  es dans la table ADMINST1.Emplois
SQL> -- Ins  rer des donn  es dans la table ADMINST1.Emplois
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (1, 'Ing  nieur logiciel', 'D  veloppement de logiciels', 80000.00, TO_DATE('2022-01-15', 'YYYY-MM-DD'), 'D  veloppement', 1);
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (2, 'Gestionnaire de projet', 'Gestion de projets informatiques', 90000.00, TO_DATE('2021-11-20', 'YYYY-MM-DD'), 'Gestion de projets', 2);
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (3, 'Sp  cialiste des ressources humaines', 'Gestion des RH', 60000.00, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 'Ressources humaines', 3);
1 ligne cr  e.

SQL>
SQL> INSERT INTO ADMINST1.Emplois (EmploiID, Titre, Description, Salaire, DateEmbauche, Departement, ResponsableID)
  2 VALUES (4, 'Analyste financier', 'Analyse des finances de l'entreprise', 75000.00, TO_DATE('2022-09-05', 'YYYY-MM-DD'), 'Finance', 4);
1 ligne cr  e.
```

```
SQL> -- Client 1
SQL> INSERT INTO CLIENTS (clientID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES (1, 'Doe', 'John', TO_DATE('1990-01-15', 'YYYY-MM-DD'), '123 Main Street', 'john.doe@email.com', '555-123-4567');
1 ligne cr  e.

SQL>
SQL> -- Client 2
SQL> INSERT INTO CLIENTS (clientID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES (2, 'Smith', 'Jane', TO_DATE('1985-07-25', 'YYYY-MM-DD'), '456 Elm Street', 'jane.smith@email.com', '555-987-6543');
1 ligne cr  e.

SQL>
SQL> -- Client 3
SQL> INSERT INTO CLIENTS (clientID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES (3, 'Dupont', 'Marie', TO_DATE('1988-05-20', 'YYYY-MM-DD'), '789 Oak Avenue', 'marie.dupont@email.com', '555-222-3333');
1 ligne cr  e.

SQL>
SQL> -- Client 4
SQL> INSERT INTO CLIENTS (clientID, Nom, Prenom, DateNaissance, Adresse, Email, Telephone)
  2 VALUES (4, 'Garcia', 'Carlos', TO_DATE('1995-11-10', 'YYYY-MM-DD'), '456 Pine Street', 'carlos.garcia@email.com', '555-777-8888');
1 ligne cr  e.
```



```

SQL> -- Produit 1
SQL> INSERT INTO PRODUIT (ProduitID, Nom, QUANTITE, prix)
  2 VALUES (1, 'Ordinateur portable', 10, 999.99);

1 ligne cr  e.

SQL>
SQL> -- Produit 2
SQL> INSERT INTO PRODUIT (ProduitID, Nom, QUANTITE, prix)
  2 VALUES (2, 'Smartphone', 20, 499.99);

1 ligne cr  e.

SQL>
SQL> -- Produit 3
SQL> INSERT INTO PRODUIT (ProduitID, Nom, QUANTITE, prix)
  2 VALUES (3, 'Tablette', 15, 299.99);

1 ligne cr  e.

SQL>
SQL> -- Produit 4
SQL> INSERT INTO PRODUIT (ProduitID, Nom, QUANTITE, prix)
  2 VALUES (4, 'Casque sans fil', 30, 79.99);

1 ligne cr  e.

SQL>

```

### **Cr  ation de la vue**

```

SQL> CREATE VIEW Vue_EmployeEmploi AS
  2 SELECT E.Nom, E.Prenom, E.Adresse, Emp.Titre, Emp.D  partement
  3 FROM Employ  s E
  4 JOIN Emplois Emp ON E.EmployeeID = Emp.ResponsableID;

Vue cr  e.

SQL>

```

### **Affectation des privil  ges au secr  taire**

```

SQL> grant select on Vue_EmployeEmploi to Rsecr;

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

SQL> grant Rsecr to SECR;

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

```

## **Teste sur secr  taire**



```
SQL> conn secr/SECR
Connecté.
SQL> select * from ADMINST1.Vue_EmployeEmploi;

NOM                                TITRE                                PRENOM                                ADRESSE
-----
DEPARTEMENT
Dupont                               Ingénieur logiciel                    Jean                                123 Rue de la Paix
Développement
Martin                               Gestionnaire de projet                 Sophie                             456 Avenue des Roses
Gestion de projets
Lefebvre                             Spécialiste des ressources humaines    Paul                               789 Boulevard du Soleil
Ressources humaines

NOM                                TITRE                                PRENOM                                ADRESSE
-----
DEPARTEMENT
Dubois                               Analyste financier                     Marie                             567 Chemin de la Lune
Finance
```

## Département Ressource humaine

### Privilèges de responsable ressource humaine

```
SQL Plus

SQL> conn ADMINST1/ADMINST1
Connecté.
SQL> grant all on Employes to RRH;

Autorisation de privilèges (GRANT) acceptée.

SQL> grant all on emplois to RRH;

Autorisation de privilèges (GRANT) acceptée.

SQL> grant RRH to RESPO_RH;

Autorisation de privilèges (GRANT) acceptée.

SQL> grant select on Vue_EmployeEmploi to RESPO_RH with grant option;

Autorisation de privilèges (GRANT) acceptée.

SQL>
```

**TESTE**

```

SQL Plus
SQL> select * from ADMINST1.emplois;

EMPLOIID TITRE
SALAIRE DATEEMBA DEPARTEMENT
RESPONSABLEID
-----
1 Ingénieur logiciel Développement de logi
ciels 80000 15/01/22 Développement
1
2 Gestionnaire de projet Gestion de projets in
formatiques 90000 20/11/21 Gestion de projets
2
3 Spécialiste des ressources humaines Gestion des RH
60000 10/03/23 Ressources humaines
3
4 Analyste financier Analyse des finances
de l'entreprise 75000 05/09/22 Finance
4

SQL> show user
USER est "RESPO_RH"
SQL>

```

```

SQL Plus
SQL> select * from ADMINST1.Vue_EmployeEmploi;

NOM ADRESSE PRENOM
TITRE
-----
DEPARTEMENT
Dupont 123 Rue de la Paix Jean
Ingénieur logiciel
Développement
Martin 456 Avenue des Roses Sophie
Gestionnaire de projet
Gestion de projets
Lefebvre 789 Boulevard du Soleil Paul
Spécialiste des ressources humaines
Ressources humaines

NOM ADRESSE PRENOM
TITRE
-----
DEPARTEMENT
Dubois 567 Chemin de la Lune Marie
Analyste financier
Finance

```

```

SQL Plus
SQL> select * from ADMINST1.Employes;

EMPLOYEEID NOM PRENOM DATENAIS ADRESSE
-----
EMAIL TELEPHONE
1 Dupont Jean 10/05/80 123 Rue de la Paix
jean.dupont@email.com 123-456-7890
2 Martin Sophie 15/08/90 456 Avenue des Roses
sophie.martin@email.com 987-654-3210
3 Lefebvre Paul 25/02/85 789 Boulevard du Soleil
paul.lefebvre@email.com 555-123-7890

EMPLOYEEID NOM PRENOM DATENAIS ADRESSE
-----
EMAIL TELEPHONE
4 Dubois Marie 03/11/95 567 Chemin de la Lune
marie.dubois@email.com 333-888-9999

```

## Accord des privilèges au vice RH

Via le user responsable des ressources humaines on accord les privilèges nécessaire au vice RH (le responsable du département contrôle les employer du département)

```
SQL> conn RESPO_RH/RERH
connecté.
SQL> grant select on ADMINST1.Vue_EmployeEmploi
  2 to RVRH;
Autorisation de privilèges (GRANT) acceptée.
```

## TESTE

```
SQL> select * from ADMINST1.Vue_EmployeEmploi;

NOM          ADRESSE          PRENOM
-----
          TITRE
-----
DEPARTEMENT
-----
Dupont      123 Rue de la Paix      Jean
            Ingénieur logiciel
Développement

Martin      456 Avenue des Roses    Sophie
            Gestionnaire de projet
Gestion de projets

Lefebvre    789 Boulevard du Soleil Paul
            Spécialiste des ressources humaines
Ressources humaines

NOM          ADRESSE          PRENOM
-----
          TITRE
-----
DEPARTEMENT
-----
Dubois      567 Chemin de la Lune    Marie
            Analyste financier
Finance

SQL> show user
USER est "VICE_RESPO_RH"
SQL>
```

## Département Achat

```
SQL Plus

SQL> show user
USER est "ADMINST1"
SQL> GRANT all ON PRODUIT TO UACHAT with grant option;
Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT all ON CLIENTS TO RACHAT ;
Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT all ON COMMANDE TO UACHAT with grant option;
Autorisation de privilèges (GRANT) acceptée.

SQL>
```

## Le caissier

Via le user responsable d'achat on accord les privilèges nécessaire au **caissier** (*le responsable du département contrôle les employé du département*)

```
SQL> conn UACHAT/UACHAT
Connecté.
SQL> GRANT SELECT ON ADMINST1.PRODUIT TO Rcaissier;

Autorisation de privilèges (GRANT) acceptée.

SQL> GRANT SELECT, INSERT ON ADMINST1.COMMANDE TO Rcaissier;

Autorisation de privilèges (GRANT) acceptée.
```

## TEST

Sur UACHAT responsable du département

### Le select

```
SQL> conn UACHAT/UACHAT
Connecté.
SQL> select * from ADMINST1.PRODUIT;
```

PRODUITID	NOM	QUANTITE	PRIX
1	Ordinateur portable	10	999,99
2	Smartphone	20	499,99
3	Tablette	15	299,99
4	Casque sans fil	30	79,99

### L'insert

```
SQL> insert into ADMINST1.PRODUIT values (5,'iphone',800,12000);

1 ligne créée.
```

### Le DELETE

```
SQL> delete from ADMINST1.PRODUIT where produitid= 5;

1 ligne supprimée.
```

### SUR LE CAISSIER

```
SQL> conn UCAISSER/UCAISSER
Connecté.
SQL> SELECT * from ADMINST1.PRODUIT;

PRODUITID NOM                                QUANTITE    PRIX
-----
1 Ordinateur portable                        10         999,99
2 Smartphone                                20         499,99
3 Tablette                                  15         299,99
4 Casque sans fil                            30          79,99

SQL> SELECT * from ADMINST1.COMMANDE;

PRODUIT_ID CLIENT_ID
-----
1          1
2          2
3          3
4          4

SQL> insert into ADMINST1.COMMANDE values(1,4);
1 ligne créée.
```

## Le client

le client visualise ses commandes propre à lui .

le client visualise la table produit pour sélectionner les produits qu'il veut commander .

on crée la vue commande chez l'administrateur « avec le check option pour limiter le client à voir seulement

```
SQL> conn sys as sysdba
Entrez le mot de passe :
Connecté.
SQL> GRANT CONNECT, RESOURCE TO CLIENT;

Autorisation de privilèges (GRANT) acceptée.

SQL> conn ADMINST1/ADMINST1
Connecté.
SQL> GRANT RCLIENT TO CLIENT;

Autorisation de privilèges (GRANT) acceptée.

SQL> CREATE OR REPLACE VIEW VueCommandes AS
 2 SELECT
 3     C.Nom AS Nom_Client,
 4     P.Nom AS Nom_Produit
 5 FROM
 6     COMMANDE CMD
 7     INNER JOIN CLIENTS C ON CMD.CLIENT_ID = C.clientID
 8     INNER JOIN PRODUIT P ON CMD.PRODUIT_ID = P.ProduitID;
```

**On donne le droit responsable département d'achat le privilege select with grant option sur la vue pour donner au client son droit de voir cet vue**

```

SQL> conn ADMINST1/ADMINST1
Connecté.
SQL> revoke SELECT ON ADMINST1.VueCommandes from Rachat;

Suppression de privilèges (REVOKE) acceptée.

SQL> GRANT SELECT ON ADMINST1.VueCommandes TO uachat with grant option;

Autorisation de privilèges (GRANT) acceptée.

SQL> conn UACHAT/UACHAT
Connecté.
SQL> GRANT SELECT ON ADMINST1.VueCommandes TO Rclient;

Autorisation de privilèges (GRANT) acceptée.

```

## Teste

```

SQL> select * from ADMINST1.VueCommandes;

NOM_CLIENT      NOM_PRODUIT
-----
Doe              Ordinateur portable
Garcia           Ordinateur portable
Smith            Smartphone
Dupont           Tablette
Garcia           Casque sans fil
SQL>

```

### 5-Les profils

Chaque profil a été créé en analysant attentivement ses droits dans le but de garantir un niveau de sécurité optimal.

**PROFIL\_ADMIN** (pour l'administrateur) :

- SESSIONS\_PER\_USER : Illimité (ou une valeur élevée si nécessaire)
- CPU\_PER\_SESSION : Illimité (ou une valeur élevée si nécessaire)
- CPU\_PER\_CALL : Illimité (ou une valeur élevée si nécessaire)
- CONNECT\_TIME : Illimité (ou une valeur élevée si nécessaire)
- IDLE\_TIME : Illimité (ou une valeur élevée si nécessaire)
- LOGICAL\_READS\_PER\_SESSION : Illimité (ou une valeur élevée si nécessaire)
- LOGICAL\_READS\_PER\_CALL : Illimité (ou une valeur élevée si nécessaire)
- PRIVATE\_SGA : Illimité (ou une valeur élevée si nécessaire)

**PROFIL\_SECR** (pour la secrétaire) :

- SESSIONS\_PER\_USER : 5
- CPU\_PER\_SESSION : 1000
- CPU\_PER\_CALL : 200
- CONNECT\_TIME : 180
- IDLE\_TIME : 30
- LOGICAL\_READS\_PER\_SESSION : 1000
- LOGICAL\_READS\_PER\_CALL : 100
- PRIVATE\_SGA : 20M

---

**PROFIL\_RH** (pour les employés du département des ressources humaines) :

- SESSIONS\_PER\_USER : 10
- CPU\_PER\_SESSION : 1500
- CPU\_PER\_CALL : 300
- CONNECT\_TIME : 240
- IDLE\_TIME : 60
- LOGICAL\_READS\_PER\_SESSION : 1500
- LOGICAL\_READS\_PER\_CALL : 150
- PRIVATE\_SGA : 30M

**PROFIL\_RACHAT** (pour les employés du département des achats) :

- SESSIONS\_PER\_USER : 5
- CPU\_PER\_SESSION : 1000
- CPU\_PER\_CALL : 200
- CONNECT\_TIME : 180
- IDLE\_TIME : 30
- LOGICAL\_READS\_PER\_SESSION : 1000
- LOGICAL\_READS\_PER\_CALL : 100
- PRIVATE\_SGA : 20M

**PROFIL\_CAISSIER** (pour les caissiers) :

- SESSIONS\_PER\_USER : 5
- CPU\_PER\_SESSION : 1000
- CPU\_PER\_CALL : 200
- CONNECT\_TIME : 180
- IDLE\_TIME : 30
- LOGICAL\_READS\_PER\_SESSION : 1000
- LOGICAL\_READS\_PER\_CALL : 100
- PRIVATE\_SGA : 20M

**PROFIL\_CLIENT** (pour les clients) :

- SESSIONS\_PER\_USER : 1
- CPU\_PER\_SESSION : 500
- CPU\_PER\_CALL : 100
- CONNECT\_TIME : 30
- IDLE\_TIME : 10
- LOGICAL\_READS\_PER\_SESSION : 500
- LOGICAL\_READS\_PER\_CALL : 50
- PRIVATE\_SGA : 10M



```
SQL Plus

SQL> alter session set "_ORACLE_SCRIPT" = true;
Session modifié.

SQL> CREATE PROFILE PROFIL_ADMIN LIMIT
2 SESSIONS_PER_USER UNLIMITED
3 CPU_PER_SESSION UNLIMITED
4 CPU_PER_CALL UNLIMITED
5 CONNECT_TIME UNLIMITED
6 IDLE_TIME UNLIMITED
7 LOGICAL_READS_PER_SESSION UNLIMITED
8 LOGICAL_READS_PER_CALL UNLIMITED
9 PRIVATE_SGA UNLIMITED;
Profil créé.

SQL> CREATE PROFILE PROFIL_SECR LIMIT
2 SESSIONS_PER_USER 5
3 CPU_PER_SESSION 1000
4 CPU_PER_CALL 200
5 CONNECT_TIME 180
6 IDLE_TIME 30
7 LOGICAL_READS_PER_SESSION 1000
8 LOGICAL_READS_PER_CALL 100
9 PRIVATE_SGA 20M;
Profil créé.

SQL>
SQL> CREATE PROFILE PROFIL_RH LIMIT
2 SESSIONS_PER_USER 10
3 CPU_PER_SESSION 1500
4 CPU_PER_CALL 300
5 CONNECT_TIME 240
6 IDLE_TIME 60
7 LOGICAL_READS_PER_SESSION 1500
8 LOGICAL_READS_PER_CALL 150
9 PRIVATE_SGA 30M;
Profil créé.

SQL>
SQL> CREATE PROFILE PROFIL_RACHAT LIMIT
2 SESSIONS_PER_USER 5
3 CPU_PER_SESSION 1000
4 CPU_PER_CALL 200
5 CONNECT_TIME 180
6 IDLE_TIME 30
7 LOGICAL_READS_PER_SESSION 1000
8 LOGICAL_READS_PER_CALL 100
9 PRIVATE_SGA 20M;
Profil créé.

SQL>
SQL> CREATE PROFILE PROFIL_CAISSEIER LIMIT
2 SESSIONS_PER_USER 5
3 CPU_PER_SESSION 1000
4 CPU_PER_CALL 200
5 CONNECT_TIME 180
6 IDLE_TIME 30
7 LOGICAL_READS_PER_SESSION 1000
8 LOGICAL_READS_PER_CALL 100
9 PRIVATE_SGA 20M;
Profil créé.

SQL>
SQL> CREATE PROFILE PROFIL_CLIENT LIMIT
2 SESSIONS_PER_USER 1
3 CPU_PER_SESSION 500
4 CPU_PER_CALL 100
5 CONNECT_TIME 30
6 IDLE_TIME 10
7 LOGICAL_READS_PER_SESSION 500
8 LOGICAL_READS_PER_CALL 50
9 PRIVATE_SGA 10M;
```

## Puis en affecte les rôles aux utilisateurs

```
SQL Plus

SQL> ALTER USER ADMINST1 PROFILE PROFIL_ADMIN;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_SECR" à l'utilisateur SECR
SQL> ALTER USER SECR PROFILE PROFIL_SECR;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_RH" à l'utilisateur RESPO_RH
SQL> ALTER USER RESPO_RH PROFILE PROFIL_RH;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_RH" à l'utilisateur VICE_RESPO_RH
SQL> ALTER USER VICE_RESPO_RH PROFILE PROFIL_RH;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_RACHAT" à l'utilisateur UACHAT
SQL> ALTER USER UACHAT PROFILE PROFIL_RACHAT;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_CAISSEIER" à l'utilisateur UCAISSEIER
SQL> ALTER USER UCAISSEIER PROFILE PROFIL_CAISSEIER;
Utilisateur modifié.

SQL>
SQL> -- Attribuer le profil "PROFIL_CLIENT" à l'utilisateur CLIENT
SQL> ALTER USER CLIENT PROFILE PROFIL_CLIENT;
Utilisateur modifié.

SQL>
```

---

## Conclusion

La sécurité basée sur des profils ainsi des rôles constitue une composante essentielle de notre système, visant à garantir un contrôle efficace des privilèges et à renforcer la protection de nos ressources. Chaque profil et rôle ont été soigneusement élaboré pour répondre aux besoins spécifiques des utilisateurs et des départements, en limitant ou en autorisant judicieusement l'accès aux ressources critiques.