Arthur Debar
Alexis Petit

UNIVERSITÉ DE NANTES

Projet TP Oracle : Contrôle d'accès

Sécurité des SI

2020/2021

M2 MIAGE

Rédigé par :
Arthur DEBAR - Alexis PETIT

Arthur Debar
Alexis Petit

Arthur Debar
Alexis Petit

1. Introduction

L'objectif de ce TP est de concevoir et de mettre en place plusieurs moyens de contrôle d'accès sur une base de données Oracle. Nous avons choisi de modéliser la gestion des salles de concert du Stéréolux.

2. Contrôle d'accès

En premier lieu nous étudions chacun des rôles et ses différentes responsabilités. Par la suite, nous détaillons le contrôle d'accès dans un tableau avant d'illustrer le tout par un schéma de base de données, puis par un schéma représentant la hiérarchie du contrôle d'accès.

Nous avons quatre rôles : *directeur, vendeur de tickets, spectateur* et *invité* ainsi que trois tables : Concert, Spectateur et Salles

a. Rôles

i. Directeur

Le directeur a pour mission la programmation des concerts du Stéréolux ainsi que la gestion des différentes salles de la structure. Pour ce faire, il a tous les droits sur les tables Concert et Salle. En revanche, le directeur n'a qu'un accès en lecture limité à la table Spectateur.

ii. Vendeur de tickets

Le vendeur de tickets a pour mission la gestion des clients (spectateurs) de l'entreprise. Ainsi, le vendeur de tickets détient tous les droits sur la table Spectateur. De plus, il peut consulter les Concerts ayant été confirmés par le directeur ainsi que les Salles du Stéréolux.

iii. Spectateur

Les spectateurs peuvent être considérés comme les clients de l'entreprise Stéréolux. En tant que tel, ils ont accès au planning des Concerts ainsi qu'au détail des Salles de l'organisation. Les spectateurs ont une vision limitée des Concerts: ils ne peuvent voir que les Concerts ayant été confirmés par le directeur et se déroulant dans moins d'un an. De plus, les spectateurs ne peuvent voir que leurs propres données dans la table Spectateur. Les spectateurs n'ont que des droits de lecture sur notre système.

iv. Invité

L'invité est le rôle le plus limité de notre système de gestion de base de données. Il peut être assimilé à un spectateur ne s'étant pas authentifié. Ainsi, il a uniquement des

droits de lecture sur les Concerts et les Salles. Comme le spectateur, il ne peut voir que les Concerts ayant été confirmés par le directeur et se déroulant dans moins d'un an.
Étant donné qu'il n'est pas authentifié, l'invité n'a pas de droit de lecture sur la table Spectateur.

b. Schéma de base de données

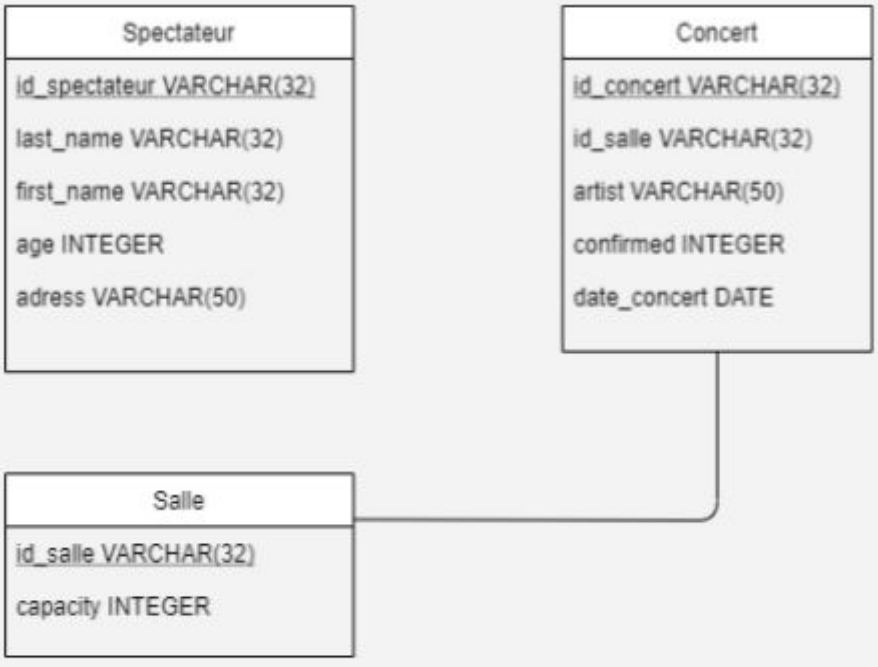

Figure 1 - Schéma représentant notre base de données

Arthur Debar
Alexis Petit

d. Tableau de contrôle d'accès

User/Table	**spectateur**	**Concert**	**Salle**
Directeur	SELECT **VUE**: (id, prenom, nom)	ALL	ALL
Vendeur de tickets	ALL	SELECT (confirmé == true) **VPD-2**	Read
Spectateur	SELECT OWN **VPD-1 & VUE**	SELECT (confirmé == true) *AND* (Date < 1 an) **VPD-2** *AND* **VPD-3**	Read
Invité	NONE	SELECT (confirmé == true) *AND* (Date < 1 an) **VPD-2** *AND* **VPD-3**	Read

e. Schéma de la hiérarchie des rôles

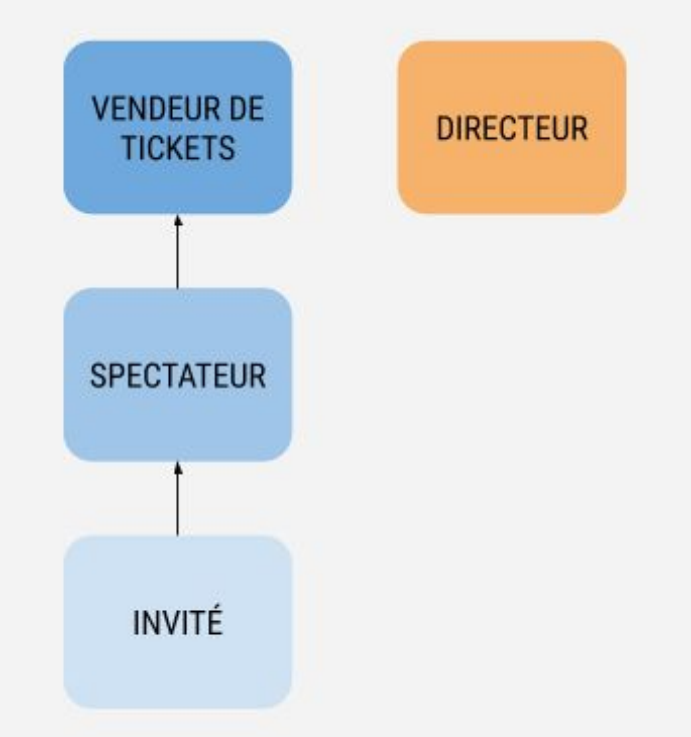

Figure 2 - Graphe de la hiérarchie des rôles

Ce schéma nous montre que le *Directeur* ne possède pas d'héritage. Cependant le rôle *vendeur de tickets* hérite de *spectateur,* qui hérite de *invité*.

Arthur Debar
Alexis Petit

3. Tables

a. Spectateur

id_spectateur	**last_name**	**first_name**	**age**	**address**
user4	Grand	Julie	28	5 rue de la poste
user5	Boular	Pascal	26	2 rue du billard
user6	Bon	Jean	35	87 avenue de la plage

b. Concert

id_concert	**id_salle**	**artist**	**confirmed**	**date_concert**
conc1	salle1	Woodkid	1	20/12/2020
conc2	salle2	Shaka Ponk	1	01/01/2021
conc3	salle1	Elephanz	1	01/05/2021
conc4	salle2	Radio Moscow	0	01/01/2023
conc5	salle1	Moby	1	01/05/2022

c. Salle

id_salle	**capacity**
salle1	2000
salle2	500

Arthur Debar
Alexis Petit

4. Démarche

a. Conception du contrôle d'accès

i. hiérarchie des rôles

Comme on le voit dans la Figure 2, nous avons mis en place une hiérarchie entre les différents rôles de notre système.
Le rôle *directeur* est isolé car il a des privilèges spécifiques sur nos trois tables.
Les rôles *invité*, *spectateur* et *vendeur de tickets* sont liés par une relation d'héritage :

- *invité* est le rôle de base avec les privilèges les plus limités. Il peut lire les tables Concert (**VPD-2** et **VPD-3**) et Salle *;*
- *spectateur* gagne un accès limité à la table Spectateur (**VPD-1**) ;
- *vendeur de tickets* a pour mission la gestion totale de la table spectateur, en plus d'avoir une vision étendue sur la table Concert (**VPD-2** uniquement).

Cet héritage a été implémenté avec le système de GRANT de rôle à rôle.

Pour réaliser notre démonstration nous avions à notre disposition sept utilisateurs. Voici la répartition de nos rôles sur ces différents utilisateurs:

User 1 : Directeur ;
User 2 : Vendeur de tickets ;
User 3 : Vendeur de tickets ;
User 4 : Spectateur ;
User 5 : Spectateur ;
User 6 : Spectateur ;
User 7 : Invité.

ii. VPD

Notre projet comporte 3 VPD :

- La **VPD-1** s'applique sur table Spectateur pour le rôle *spectateur*, celle-ci permet de retourner uniquement les informations de l'utilisateur qui s'est connecté ;

- La **VPD-2** s'applique sur le table Concert pour les rôles *vendeur de tickets, spectateur* et *invité*. Elle permet de sélectionner uniquement les concerts confirmés par le *directeur* ;

- La **VPD-3** s'applique aussi sur la table Concert pour les rôles *spectateur* et *invité*, elle permet de ne retourner que les concerts se déroulant dans moins d'un an.

Les VPD-2 et VPD-3 s'appliquent en fonction de nos rôles. Les rôles *invité* et *spectateur* cumulent les deux VPD tandis que le rôle *vendeur de ticket* n'est contraint que par la VPD-2.

iii. Vue

Notre projet comporte une VUE sur la table Spectateur pour le rôle *directeur*. Nous avons pensé que pour des questions de RGPD le directeur ne devait avoir accès qu'à des informations limitées sur les spectateurs. Ainsi, notre VUE permet de retourner uniquement les id, nom et prénom de nos spectateurs.

b. Organisation du travail

Afin de faciliter le travail en groupe nous avons partagé notre code avec le gestionnaire de version GIT.

Avant de commencer la rédaction des scripts, nous nous sommes concentrés sur la conception du système. En premier lieu nous avons déterminé quelles tables intégrer et quels rôles attribuer. Par la suite, nous avons pu déterminer différentes contraintes à appliquer sur ces tables selon les différents rôles. Cette étape nous a permis de mettre en exergue les VPD et la Vue à mettre en place, ainsi que la hiérarchie à appliquer entre les différents rôles du système (voir partie 4.a).

Afin de mettre en place notre base de données nous avons commencé par créer le script permettant d'initialiser nos différentes tables et d'insérer des données. Nous avons aussi réalisé un script permettant de supprimer toutes les données de notre base afin de repartir sur une base propre à chaque lancement du script.
Ensuite, nous avons intégré au script d'initialisation la création des différents rôles et leur attribution aux utilisateurs fournis.
Une fois les différents rôles attribués, nous avons pu créer le contexte auquel s'attacheront les futures VPD.
Concernant les VPD, nous avons commencé par implémenter la VPD-2. Une fois le problème détaillé en 5. résolu, nous avons pu rédiger le script de démonstration du rôle *vendeur de tickets* et exécuter notre code avec succès.
Pour finir, nous nous sommes ensuite réparti le travail pour gagner du temps. Alexis s'est occupé de créer les VPD-1 et VPD-3 ainsi que la VUE sur la table Directeur. Arthur s'est concentré sur la partie démonstration en créant les différents scripts pour chaque rôle utilisateurs.

5. Difficultés rencontrées

La principale difficulté que nous avons rencontrée a émergé lorsque nous avons voulu tester nos VPD. En effet, lors de l'exécution de notre première VPD nous sommes restés bloqués car nous étions confrontés à l'erreur suivante : `ORA-00942: table or view does not exist`. Ici, l'erreur nous indique que la table dba_role_privs n'existe pas. Cependant, lorsque nous faisions un SELECT sur celle-ci nous avions bien en retour les différents tuples de cette table.

Ce problème qui touchait tous les groupes a finalement été résolu par l'intervention de Patricia Serrano Alvarado. En effet, nous n'avions pas les droits suffisants sur une table particulière (dba_role_privs) dans la base de données Oracle. Suite à sa résolution nous avons pu avancer et terminer le projet.

L'autre difficulté fut de nous adapter afin de travailler à distance avec le contexte du COVID-19. En effet, comme nous ne pouvions nous rendre à l'université nous devions nous connecter à distance sur nos machines pour pouvoir accéder au CIE. Cependant cet accès n'était pas continu (notamment le week-end) et nous n'avions que certaines plages horaires pour nous y connecter, ce qui a rendu l'organisation plus compliquée.

Conclusion

Ce TP nous a permis d'approfondir nos connaissances sur les bases de données Oracle en découvrant différentes solutions de contrôle d'accès. En effet, nous n'avions jamais eu l'occasion d'intégrer des rôles dans une base de données. De plus, les VPD sont une notion qui nous semble être avancée et que nous avons finalement pu comprendre et appliquer dans notre projet. Les difficultés que nous avons rencontrées nous ont aussi permis de gagner en maturité sur ce type de projet. Il sera intéressant par la suite d'approfondir la gestion des droits utilisateurs/administrateurs d'une base de données Oracle.