Contrôle d'accès dans les SGBD

Patricia Serrano Alvarado
Université de Nantes
Laboratoire LINA

Introduction

- Sécurité
  - Confidentialité
    - L'information ne doit pas être dévoilée aux utilisateurs non-autorisés. Uniquement les utilisateurs autorisés doivent modifier les données
  - Intégrité
    - Les modifications doivent respecter les contraintes d'intégrité des données ainsi que les propriétés transactionnelles
  - Disponibilité
    - Les données doivent être toujours disponibles pour les utilisateurs autorisés. Les données doivent être tolérantes aux pannes et l'accès doit être performant

Techniques pour assurer la confidentialité des données

- Control d'accès (RBAC, DAC, MAC)
- Vues
- Cryptographie
- Bases de données statistiques et résumées

Intégrité

- Une base de données est cohérente si elle respecte les **propriétés transactionnelles**
  - Atomicité. Toutes les modifications faites par une transaction sont validées ou aucune
  - Cohérence. Les contraintes d'intégrité doivent être vérifiées et respectées
  - Isolation. Chaque transaction s'exécute « comme » si elle est la seule à être exécutée sur le système
  - Durabilité. Une fois qu'une transaction est validée, toutes ses modifications doivent persister malgré des éventuelles pannes

Intégrité

- Les propriétés transactionnelles sont assurées par le gestionnaire des transactions
  - Atomicité et durabilité. Protocole de validation (e.g., validation à deux phases 2PC)
  - Cohérence. Contraintes d'intégrité sémantiques
  - Isolation. Protocoles de control de concurrence (e.g., verrouillage à deux phases 2PL)

Intégrité

- Contraintes d'intégrité
  - Règles qui représentent les propriétés d'une application
    - Contraintes structurelles : ex. clés primaires
    - Contraintes comportementales : ex. rang de valeurs pour un attribut,
- Une base de données doit assurer ces contraintes
- Le gestionnaire de la base de données assure les contraintes
  - Activité complexe et coûteuse lorsque la base est répartie mais aussi lorsque, lors des mises à jour, vérifier les contraintes demande l'accès à un grand nombre de données

Gestion des contraintes d'intégrité

- Dans l'idéal la gestion de contraintes devrait
  - Limiter le nombre de contraintes à assurer
  - Réduire le nombre de données accédées lors d'une mise à jour
  - Avoir des stratégies pour détecter des inconsistances entre les contraintes afin d'éviter de défaire les mises à jour
  - Réaliser le moins de control en temps réel

Disponibilité

- Disponibilité de données malgré les pannes et les nombreuses demandes d'accès
  - Accès efficace et rapide qui passe à l'échelle
    - Eviter les goulots d'étranglement
    - Faciliter l'accès avec de données plus « proches » de l'utilisateur
  - Type de pannes
    - Système. La mémoire centrale est perdue
    - Disque. La mémoire secondaire est endommagée
    - Transactionnelle. Erreur dans l'exécution des transactions

Solutions pour la disponibilité

- Accès efficace
  - Duplication de données
  - Systèmes pair-à-pair
- Données disponibles malgré les pannes
  - Disque : sauvegardes périodiques, point de reprise
    - Sauvegarde. Copie cohérente d'une base de données effectuée périodiquement alors que cette base est dans un état cohérent
    - Point de reprise. Etat d'avancement du système sauvegardé sur mémoires secondaires à partir duquel il est possible de repartir après un arrêt
  - Transactionnelle et système: journaux des images avant et après utilisés dans les procédures de reprise
    - Ex. Procédure de reprise à chaud. A partir d'un point de reprise, les transactions non validées sont défaites et les validées sont ré exécutées.

Contrôle d'accès

DAC (Discretionary Access Control)
MAC (Mandatory Access Control)
RBAC (Role-Based Access Control)

Control d'accès dans les SGBD

- Discretionary Access Control (DAC)
  - Basé sur des privilèges ou droits d'accès
  - Gestion de privilèges par les sujets (processus des utilisateurs)
- Mandatory Access Control (MAC)
  - Basé sur des politiques du système
  - Gestion de politiques par le système (pas par les utilisateurs)
- Role-based Access Control (RBAC)
  - Basé sur des rôles
  - Gestion de rôles et leur privilèges par une unité centrale

DAC

- Consiste à attribuer aux sujets des privilèges des opérations sur les objets et à les vérifier lors de l'accès
- Un privilège permet à un sujet l'accès à une donnée (e.g., table, vue, objet, procédure)
  - SELECT : privilège pour lire toutes le colonnes de la donnée ainsi que les colonnes insérées plus tard avec ALTER TABLE
  - INSERT/UPDATE : privilège pour insérer/modifier des lignes
  - DELETE : privilège pour supprimer des lignes
  - REFERENCES : privilège pour définir des clés étrangères (référencier d' autres tables ou vues)
  - EXECUTE : privilège pour exécuter une fonction/procédure

DAC

- Gestion de privilèges
  - Les privilèges peuvent être partagés et révoqués (commandes GRANT et REVOKE)
  - Si un privilège a été donné avec GRANT OPTION, le nouveau sujet peut également partager le privilège
  - Le sujet qui crée une table automatiquement a tous les privilèges sur celle-ci
  - La commande REVOKE peut être utilisée pour annuler un privilège ou l'option GRANT OPTION
  - REVOKE peut être utilisé avec CASCADE

Exemples

- Considérez les tables suivantes et les utilisateurs Joe, Michel, Lea, Yuppy, Bill, Eric, Guppy, Art, Bob, etc.

Sailors	Boats	Reservs
• sid: integer • sname: string • rating: integer • age: real	• bid: integer • bname: string • color: string	• sid: integer • bid: integer • day: date

Exemples

- L'utilisateur Joe crée les tables (Boats, Reservs et Sailors) et exécute les commandes :
  - GRANT INSERT, DELETE ON Reservs TO Yuppy WITH GRANT OPTION
  - GRANT SELECT ON Reservs TO Michel
  - GRANT SELECT ON Sailors TO Michel WITH GRANT OPTION
  - GRANT INSERT (sid,sname,age) ON Sailors TO Michel
  - GRANT REFERENCES (bid) ON Boats TO Bill

- Michel exécute :
  - CREATE VIEW YoungSailors (sid, age, rating)
    AS SELECT sid, age, rating
    FROM Sailors
    WHERE age<18
  - GRANT SELECT ON YoungSailors TO Eric, Guppy

- Peut Michel faire la même chose avec Reservs ?

Exemples

- Si Joe ajoute la colonne « address » à Sailors, peut Michel insérer des valeurs à la nouvelle colonne ?
- Si Joe ajoute la colonne « employe » à Reservs, peut Yuppy insérer des valeurs à la nouvelle colonne ?
- Peut Bill créer une table Reservs de la manière suivante ?
  - CREATE TABLE Reservs (sid INTEGER,
    bid INTEGER,
    day DATE,
    PRIMARY KEY (bid,day),
    FOREIGN KEY (sid) REFERENCES Sailors,
    FOREIGN KEY (bid) REFERENCES Boats)
- Qu'est-ce qui se passe si Joe exécute :
  - REVOKE REFERENCES ON Boats FROM Bill

Exemples

- Considérer les commandes suivantes :
- Joe (qui a crée Sailors) exécute :
  - GRANT SELECT ON Sailors TO Art WITH GRANT OPTION
- Art exécute :
  - GRANT SELECT ON Sailors TO Bob WITH GRANT OPTION
- Qu'est-ce qui se passe si Joe exécute :
  - REVOKE SELECT ON Sailors FROM Art CASCADE

  ou
  - REVOKE GRANT OPTION FOR SELECT ON Sailors FROM Art CASCADE

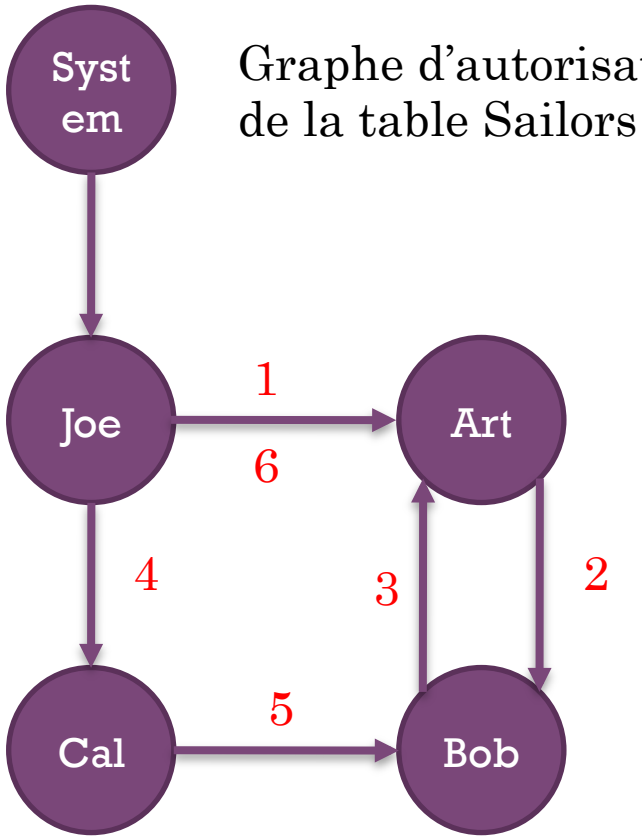
- Avec l'option RESTRICT la commande REVOKE est rejetée si en plus de l'utilisateur spécifié d' autres utilisateurs vont voir ses privilèges affectés

Graphes d'autorisation

- Les effets des commandes GRANT et REVOKE peuvent être décrits dans un graphe d'autorisations
  - Un graphe concerne une seule donnée
  - Les nœuds sont les utilisateurs
  - Les arcs indiquent comment les privilèges sont accordés

Exemple de graphe d'autorisation

- **Joe exécute :**
  1. GRANT SELECT ON Sailors TO Art WITH GRANT OPTION
- **Art exécute :**
  2. GRANT SELECT ON Sailors TO Bob WITH GRANT OPTION
- **Bob exécute**
  3. GRANT SELECT ON Sailors TO Art WITH GRANT OPTION
- **Joe exécute**
  4. GRANT SELECT ON Sailors TO Cal WITH GRANT OPTION
- **Cal exécute**
  5. GRANT SELECT ON Sailors TO Bob WITH GRANT OPTION
- **Joe exécute**
  6. REVOKE SELECT ON Sailors FROM Art CASCADE



Graphe d'autorisation de la table Sailors

Privilèges sur les vues

- Gestion de privilèges sur les vues
  - L'utilisateur qui crée une vue doit préalablement avoir au moins le privilège de SELECT sur les données concernées (tables/vues utilisées pour la création de la vue)
  - Si la vue peut être mise à jour, l'utilisateur doit également avoir le privilège de INSERT, UPDATE et/ou DELETE sur les données de base
  - Du même avec GRANT OPTION

Privilèges sur les vues

- Les privilèges du créateur d'une vue dépendent de ses privilèges sur les tables de base
  - Si le créateur perd un privilège obtenu avec l'option GRANT OPTION les utilisateurs auxquels il a donné le privilège le perdent également
  - Une vue peut être supprimée si son créateur perd le privilège de sélection d'une des tables de base de la vue
  - Si le créateur obtient plus de privilèges sur les tables de base d'une vue alors il obtient automatiquement les mêmes privilèges sur la vue

Vues et confidentialité - cas d'étude

Sailors	Boats	Reservs
• sid: integer • sname: string • rating: integer • age: real	• bid: integer • bname: string • color: string	• sid: integer • bid: integer • day: date

- L'utilisateur Joe crée les tables (Boats, Reservs et Sailors). Qu'est-ce qu'il doit faire pour permettre :
  - La lecture et l'écriture des marins jeunes à Michel
  - La lecture et l'écriture des marins âgés à Bill
  - La lecture de son information à chaque marin
- Joe doit exécuter :
  - Création des vues YoungSailors et OldSailors
  - Donner les droits de lecture et d'écriture à Michel et Bill
  - Création des vues de chaque marin et donner les droits de lecture à chacun (ex dans un déclencheur)

MAC

- Le control basé sur DAC a une limitation principal qui est la possibilité d'avoir de chevaux de Troie
  - L'utilisateur A a accès aux données R et S
  - L'utilisateur B a accès uniquement à S
  - Si B trafique une application utilisée par A et il écrit les données de R sur S alors B peut lire des données non autorisées sans violer les règles de DAC
- MAC a été proposé pour palier ce type de problèmes
- MAC consiste à associer un niveau d'autorisation aux sujets et aux données

MAC

- Le modèle MAC de Bell-LaPadula
  - Objets
    - Tables, vues, lignes, colonnes, etc.
  - Sujets
    - Utilisateurs, programmes
  - Classes de sécurité
    - Organisées dans un ordre partiel de la plus sécurisée à la moins sécurisée (i.e., *lattice* ou treillis)
    - Ex. 4 clases : *top secret* (TS), *secret* (S), *confidential* (C) et *unclassified* (U) où TS>S>C>U où TS est le plus sensible et U le moins
  - Autorisations (*clearance*)
  - Chaque objet appartient à une classe de sécurité
  - Chaque sujet a une autorisation pour une clase de sécurité

MAC

Les deux règles de Bell-LaPadula

1. Propriété de sécurité simple ou ***no read up***
   - Un sujet S peut lire l'objet O uniquement si

     classe(S)>= classe(O)

     Ex. un sujet avec autorisation S ne peut pas lire une donnée avec autorisation TS

2. Propriété *- ou ***no write down***
   - Un sujet S peut écrire sur un objet O si

     classe(S)<= classe(O)

     Ex. un sujet avec autorisation S peut écrire sur les objets avec autorisation S ou TS

Classification multi-niveaux

- Le MAC dans le modèle relationnel et les tables « multi-niveaux »
- Granularité de la classification (classes de sécurité)
  - Tables
  - Lignes
  - Colonnes
  - Valeur d' une colonne
- Utilisateurs avec différentes autorisations perçoivent différemment une même table
- Voici une table avec une classification par ligne :

bid (PK)	bname	color	classe de sécurité
101	Salsa	Red	S
102	Pinto	Brown	C

La table Boats

Classification multi-niveaux

- Table avec une classification par colonne
  - Chaque colonne a en plus son niveau de sécurité
  - Les niveaux de sécurité incrémentent la taille de la base

sid (PK)	SL1	sname	SL2	rating	SL3	age	SL4
AA	**C**	Tim	C	Prof essional	C	45	C
BB	**C**	Bill	C	NULL	S	NULL	S
CC	**S**	Michel	S	Amateur	S	50	S

La table Sailors

- Un utilisateur perçoit uniquement les attributs qui lui sont autorisés, pour les autres des valeurs nulles seront insérées

Poly-instanciation

- Problème de la poly-instanciation avec MAC

bid (PK)	bname	color	classe de sécurité
101	Salsa	Red	S
102	Pinto	Brown	C

La table Boats

- Si l'utilisateur U avec autorisation C veut insérer le tuple <101, Picante, Scarlet, C> soit
  - L'insertion est faite et 2 lignes auront comme clé 101
  - L'insertion est rejetée et S déduit qu'il existe un autre bateau avec comme clé 101 ce qui viole la confidentialité des objets de classe supérieure
  - Solution : les classes de sécurité font partie de la clé

RBAC

- DAC et MAC ne sont pas applicables à des systèmes complexes avec des centaines/milliers d'utilisateurs
  - Inconvénients de DAC
    - Approche basée sur l'attribution de droits par sujet (utilisateur) donc difficulté pour gérer la dynamicité des privilèges
      - Sorties du système (entreprise)
      - Modification des droits (changement de responsabilités)
  - Inconvénients de MAC
    - Approche basée sur une organisation hiérarchique des privilèges
      - Difficilement une grande organisation peut organiser ses privilèges hiérarchiquement

RBAC

- Proposé pour des systèmes très grands avec un important nombre d'utilisateurs et de données
- Contrôle d'accès à base de rôles
  - Modèle dans lequel les décisions d'accès dépendent du rôle auquel l'utilisateur est attaché
- Concepts de base
  - Utilisateur (pas de processus)
  - Privilège. Concerne un droit (opération) quelconque sur une donnée, plusieurs droits sur une donnée ou plusieurs droits sur plusieurs données
  - Rôles. Entité déterminant une activité d'entreprise (comptable, chef de projet, chef de département, cassier, etc.)
- Deux approches : modèles ANSI et *role graphe*

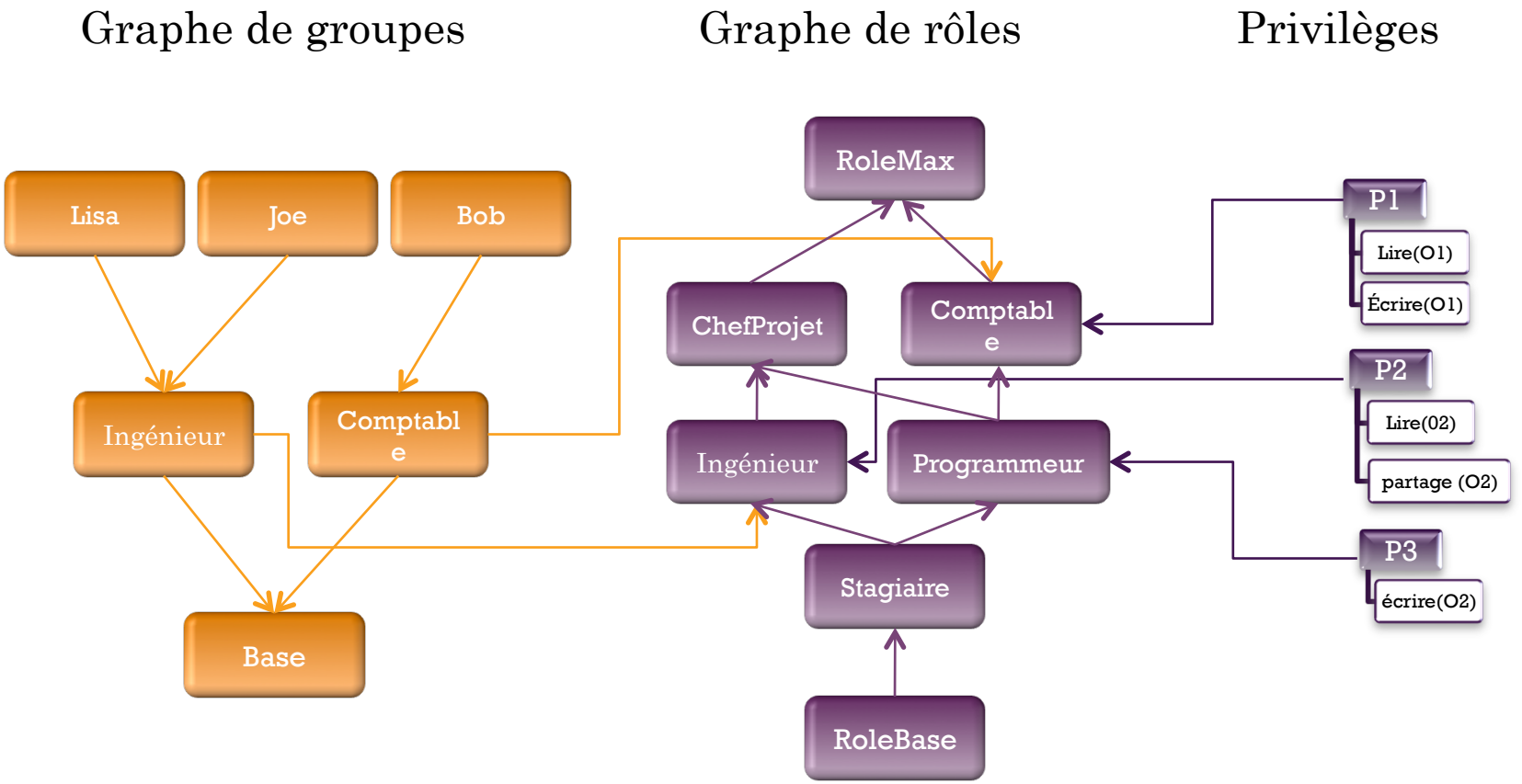
Les rôles

- Sont définis par une autorité centrale
- Sont identifiés par un nom unique
- Peuvent être organisés en une hiérarchie ou un graphe. Attention aux
  - Cycles (redondance)
  - Conflits d'intérêt
- Sont attribués aux utilisateurs ou groups d'utilisateurs
  - Mapping rôle-utilisateur
  - Plusieurs rôles peuvent être attribués à un utilisateur
  - Plusieurs utilisateurs peuvent partager plusieurs rôles
- Facilité de gestion rôle-utilisateur

Les groupes d'utilisateurs et les rôles

- Groupes d'utilisateurs
  - Facilite la gestion des utilisateurs
  - Les membres du group peuvent changer fréquemment
  - La gestion des groupes peut se faire par le gestionnaire des ressources humaines
- Rôles
  - les rôles sont définis avant la mise en place du système
  - Les privilèges attribués aux rôles varient très peu
  - La gestion des rôles doit se faire par un utilisateur de confiance

Composants du modèle basé sur les rôles



Propriétés du graphe de rôles

- Héritage
  - r1→r2 indique que r1 est junior de r2 et que les privilèges de r1 sont hérités à r2
  - privilèges(r1) ⊆ privilèges(r2)
- Privilèges directs
  - Privilèges attribués directement au rôle par l'administrateur du graphe
- Privilèges effectifs
  - Privilèges directs plus privilèges hérités
- Graphe acyclique
- Pas de redondance de privilèges
  - Un arc r1→r2 doit être ajouté si privilèges(r1) ⊂ privilèges(r2)

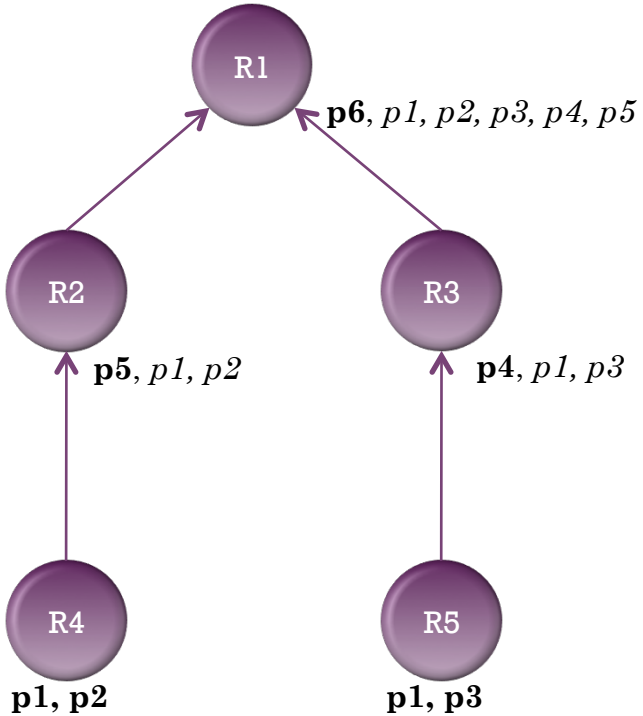
Gestion du graphe de rôles

- roleAddition
  - Si pas de cycle, le rôle est ajouté
- permitionAddition
  - Un nouveau privilège est ajouté à un rôle
- permitionDeletion
  - Suppression d'un privilège d'un rôle
- roleDeletion
  - Suppression d'un rôle
- edgeInsertion
  - Insertion d'un arc si pas de cycle
- edgeDeletion
  - Suppression d'un arc si pas de cycle

- Dans toutes les fonctions, si nécessaire, les privilèges/arcs sont réorganisés pour éviter les redondances

Exemple 1 de gestion de graphe de rôles

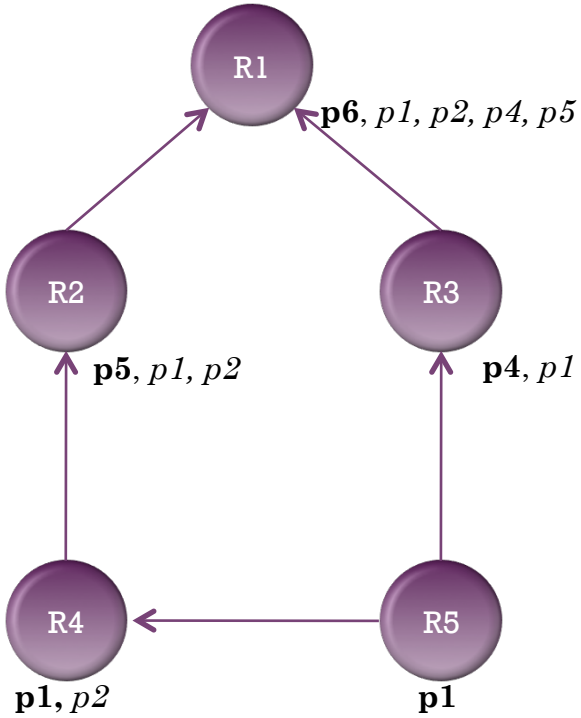
Graphe A



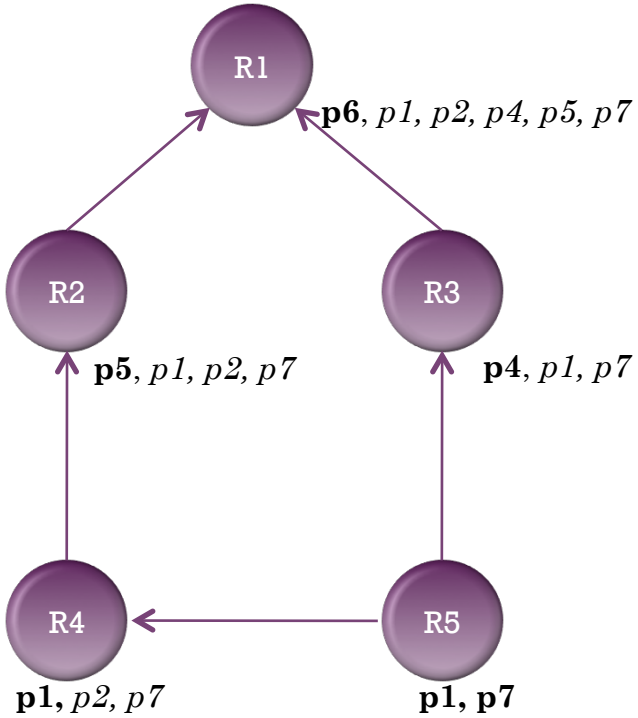
Pas de cycle
Pas de redondance de privilèges i.e., ¬(privilèges(r1) ⊂ privilèges(r2))

Exemple 2 de gestion de graphe de rôles

Graphe A, la suite

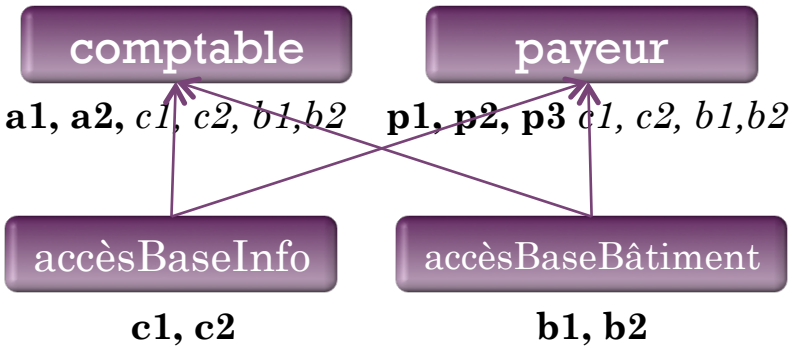


Graphe A avec p7 ajouté à R5

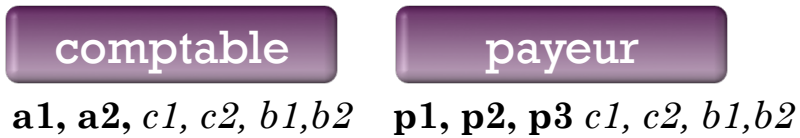


Conseils pour définir une hiérarchie de rôles

- Ne pas faire de la hiérarchie de rôles un miroir de la hiérarchie de l'entreprise
- Surveiller les privilèges des rôles
- Considérer les groupes d'utilisateurs
- Considérer de rôles abstraits



Avec des rôles abstraits



Sans des rôles abstraits

Conclusion

- Le control d'accès est vital pour assurer la confidentialité des données
- DAC, MAC, RBAC : les trois grandes approches
  - Chacun a ses avantages et désavantages
  - RBAC se révèle le plus adapté aux systèmes actuels

Bibliographie

- Security, Privacy, and Trust in Data Management. Milan Petkovic, Willem Jonker Eds. 2007.
- Database Management Systems. Raghu Ramakrishnan, Johannes Gehrke. International 3rd Edition, 2003.

Les rôles dans Oracle

Les rôles dans Oracle

- Privilège
  - Droit d'exécuter une sentence SQL ou d'accéder les objets (tables, vues, objets, etc.) d'un autre utilisateur
  - Un privilège représente le couple (objet, privilège)
  - 173 privilèges dans Oracle 10g (SYSTEM_PRIVILEGE_MAP)
- Rôle
  - Ensemble de privilèges qui peut être attribué aux utilisateurs ou aux rôles

Types de privilèges

- Privilèges objets
  - Privilèges permettant la gestion des objets crées par les utilisateurs
  - CREATE, SELECT, INSERT, etc.
- Privilèges système
  - Privilèges permettant la gestion du système de bases de données
  - Ne sont pas liés à un objet ou schéma
  - CREATE SESSION, CREATE ROLE, ALTER ROLE, etc.
- Rôles
  - Peuvent contenir de privilèges objets et système

Privilèges objets

Objet	Privilège
Tables	select, insert, update, delete, alter, debug, flashback, on commit refresh, query rewrite, references, all, index
Views	select, insert, update, delete, under, references, flashback, debug
Sequence	alter, select
Packeges, Procedures, Functions (Java classes, sources...)	execute, debug
Materialized Views	delete, flashback, insert, select, update
Directories	read, write
Libraries	execute
User defined types	execute, debug, under
Operators	execute
Indextypes	execute

Privilèges sur les objets

Object Privilege	Table	View	Sequence	Procedures, Functions, Packages	Materialized View	Directory	Library	User-defined Type	Operator	Index-type
ALTER	X		X							
DELETE	X	X			X[b]					
EXECUTE				X[c]			X[c]	X[c]	X[c]	X[c]
DEBUG	X	X		X				X		
FLASHBACK	X	X			X					
INDEX	X									
INSERT	X	X			X[b]					
ON COMMIT REFRESH	X									
QUERY REWRITE	X									
READ						X				
REFERENCES	X	X								
SELECT	X	X	X		X					
UNDER		X						X		
UPDATE	X	X			X[b]					
WRITE						X				

Privilèges système

- Définis par Oracle et pas modifiables par les utilisateurs
- Peuvent être attribués uniquement par les administrateurs du système ayant comme privilèges
  - Le privilège à attribuer avec l'option ADMIN OPTION
  - Le privilège général GRANT ANY PRIVILEGE
- Les privilèges sur les objets ne peuvent pas être attribués avec des privilèges systèmes ou rôles dans la même sentence GRANT

L'option ADMIN OPTION

- Utilisable avec les rôles et les privilèges système
- Semblable à WITH GRANT OPTION des privilèges objets
- Le receveur peut
  - Attribuer ou révoquer le rôle ou le privilège système
  - Attribuer le rôle ou le privilège système avec l'option ADMIN OPTION
  - Modifier (ALTER) ou supprimer le rôle
- Pour annuler l'option, il faut révoquer le privilège
- Exemples
  - GRANT myRole TO alice WITH ADMIN OPTION;
  - REVOKE myRole FROM alice;

Avantages de l'utilisation de rôles

- Gestion de privilèges réduite
  - Au lieu d'attribuer un ensemble de privilèges aux utilisateurs, un par un, les privilèges sont attribués à un rôle et uniquement le rôle est attribué aux utilisateurs
- Gestion dynamique des privilèges
  - Si les privilèges d'un groupe change, uniquement le rôle sera modifié
- Surveillance plus simple
  - Plus facile de vérifier les privilèges des utilisateurs

Utilisation des rôles

- Pour la gestion des privilèges d'un group d'utilisateurs
- Pour la gestion des privilèges d'une application de base de données
  - Attribution à un rôle des privilèges nécessaires à l'exécution d'une application
  - Le rôle d'application crée sera attribué aux utilisateurs de l'application

Rôles prédéfinis d'Oracle

- Définis automatiquement lors de l'installation d'Oracle
- Aident à la gestion de la base de données
- Peuvent être gérés comme les rôles définis par un utilisateur
- Exemples
  - CONNECT : CREATE VIEW, CREATE TABLE, ALTER SESSION, CREATE CLUSTER, CREATE SESSION, CREATE SYNONYM, CREATE SEQUENCE, CREATE DATABASE LINK
  - RESOURCE : CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE

Gestion de rôles

- Création/suppression
  - Un rôle peut être crée et supprimé
- Sécurisation
  - Les rôles peuvent être protégés par un mot de passe pour éviter que n'importe qui les ajoute/supprime/attribue de privilèges
- Attribution
  - Un rôle peut être attribué aux utilisateurs ou aux rôles
  - On peut attribuer des privilèges aux rôles

Création/suppression de rôles

- Création de rôles
  - Uniquement par des utilisateurs
    - Ayant le privilège CREATE ROLE
- Modification de rôles
  - Par les utilisateurs
    - Ayant eu attribué le rôle avec l'option ADMIN OPTION
    - Ayant le privilège ALTER ANY ROLE
- Suppression de rôles
  - Par les utilisateurs
    - Ayant créé le rôle
    - Ayant eu attribué le rôle avec l'option ADMIN OPTION
    - Ayant le privilège DROP ANY ROLE

Création/suppression de rôles

- Création et sécurisation
  - Les rôles peuvent être crées avec un mot de passe
    - CREATE ROLE myRole;
    - CREATE ROLE myRole IDENTIFIED BY my_pwd;
  - Un mot de passe peut également être attribué à un rôle existant
    - ALTER ROLE myRole IDENTIFIED BY my_pwd;
- Suppression
  - DROP ROLE myRole;

Attribution de privilèges à un rôle

- L'attribution peut être fait par les utilisateurs
  - Ayant créé le rôle
  - Ayant eu le rôle attribué (avec ou sans l'option ADMIN OPTION)
  - Ayant le privilège système GRANT ANY OBJECT PRIVILEGE (pour les privilèges objets)
  - Ayant le privilège système GRANT ANY PRIVILEGE (pour les privilèges système)
  - Exemples
    - GRANT SELECT ON myTable TO myRole;
    - GRANT SELECT ON myTable TO myRole1, myRole2;
    - GRANT SELECT, UPDATE ON myTable TO myRole;

Suppression de privilèges à un rôle

- La suppression peut être faite par des utilisateurs
  - Ayant préalablement attribué le privilège au rôle
  - Ayant le privilège système GRANT ANY OBJECT PRIVILEGE
  - Exemples
    - REVOKE SELECT ON myTable FROM myRole;
    - REVOKE ALL ON myTable FROM myRole;

Attribution de rôles

- Les rôles peuvent être attribués uniquement par les utilisateurs
  - Ayant eu le rôle attribué avec l'option ADMIN OPTION
  - Le privilège général GRANT ANY ROLE
  - Exemples
    - Attribution de rôle à rôle
      - GRANT myRole TO myRole2;
      - GRANT myRole, myRole3 TO myRole2, myRole4;
      - Interdiction de cycles ! -> ~~GRANT myRole2 TO myRole;~~
    - Attribution de rôle aux utilisateurs
      - GRANT myRole TO bob, alice;
      - REVOKE myrole TO bob;

Révocation de rôles

- Un rôle peut être révoqué par les utilisateurs
  - Ayant attribué le rôle
  - Ayant eu le rôle attribué avec l'option ADMIN OPTION
  - Ayant le privilège système GRANT ANY ROLE
  - Exemples
    - REVOKE myRole TO bob;
    - REVOKE myRole TO medecins;

Le privilège GRANT ANY OBJECT PRIVILEGE

- Permet d'attribuer/révoquer des privilèges à la place d'autres utilisateurs
  - Trois utilisateurs : A, B, C
  - A est un administrateur de la BD qui possède le privilège GRANT ANY OBJECT PRIVILEGE
  - B a créé la table table_b
  - L'utilisateur A exécute

    GRANT SELECT ON B.table_b TO C;

  - Cela est comme si B a attribué le droit directement à C
  - La même opération peut être faite avec REVOKE

Où trouver l'information sur les privilèges ?

- Dans les dictionnaires d'Oracle
  - Un dictionnaire est une vue gérée automatiquement par Oracle
- Les dictionnaires portent sur tout type d'information liée à la BD
  - Utilisateurs
  - Sessions
  - Les objets
  - Les privilèges
  - Les rôles
  - Etc.

Quelques dictionnaires

- Privilèges
  - USER_COL_PRIVS : montre les colonnes des objets sur lesquelles l'utilisateur actuel est le owner/grantor/grantee
  - USER_TAB_PRIVS : montre les privilèges sur les objets où l'utilisateur est le grantee
- Rôles
  - DBA_ROLES : montre tous les rôles du système
  - USER_ROLE_PRIVS : montre les rôles attribués à l'utilisateur
- Système
  - USER_SYS_PRIVS : montre les privilèges système attribués à l'utilisateur
  - SYSTEM_PRIVILEGE_MAP : montre tous les privilèges du système

Table 4–7 Views That Display Grant Information about Privileges and Roles

View	Description
ALL_COL_PRIVS	Describes all column object grants for which the current user or PUBLIC is the object owner, grantor, or grantee
ALL_COL_PRIVS_MADE	Lists column object grants for which the current user is object owner or grantor.
ALL_COL_PRIVS_RECD	Describes column object grants for which the current user or PUBLIC is the grantee
ALL_TAB_PRIVS	Lists the grants on objects where the user or PUBLIC is the grantee
ALL_TAB_PRIVS_MADE	Lists the all object grants made by the current user or made on the objects owned by the current user.
ALL_TAB_PRIVS_RECD	Lists object grants for which the user or PUBLIC is the grantee
DBA_COL_PRIVS	Describes all column object grants in the database
DBA_TAB_PRIVS	Lists all grants on all objects in the database
DBA_ROLES	This view lists all roles that exist in the database, including secure application roles
DBA_ROLE_PRIVS	Lists roles granted to users and roles
DBA_SYS_PRIVS	Lists system privileges granted to users and roles
ROLE_ROLE_PRIVS	This view describes roles granted to other roles. Information is provided only about roles to which the user has access.

View	Description
ROLE_SYS_PRIVS	This view contains information about system privileges granted to roles. Information is provided only about roles to which the user has access.
ROLE_TAB_PRIVS	This view contains information about object privileges granted to roles. Information is provided only about roles to which the user has access.
USER_COL_PRIVS	Describes column object grants for which the current user is the object owner, grantor, or grantee
USER_COL_PRIVS_MADE	Describes column object grants for which the current user is the grantor
USER_COL_PRIVS_RECD	Describes column object grants for which the current user is the grantee
USER_ROLE_PRIVS	Lists roles granted to the current user
USER_TAB_PRIVS	Lists grants on all objects where the current user is the grantee
USER_SYS_PRIVS	Lists system privileges granted to the current user
USER_TAB_PRIVS_MADE	Lists grants on all objects owned by the current user
USER_TAB_PRIVS_RECD	Lists object grants for which the current user is the grantee
SESSION_PRIVS	Lists the privileges that are currently enabled for the user
SESSION_ROLES	Lists the roles that are currently enabled to the user

D’autres dictionnaires

- SESSION_ROLES
  - Tous les rôles actifs
- USER_SOURCE
  - Le code des procédures appartenant à l’utilisateur
- ALL_SOURCE
  - Le code des procédures appartenant à l’utilisateur ou à ceux auxquels il a accès
- DBA_SOURCE
  - Toutes les procédures de la BD
- USER_CATALOG
  - Information sur les tables, vues, séquences et synonymes de l’utilisateur
- USER_OBJECTS
  - Tout type d’ objet Oracle (*clusters, database links, directories, functions, indexes, libraries, packages, java classes, abstract datatypes, resource plans, sequences, synonyms, tables, triggers, materialized views, LOBs, and views*)

Conclusion

- DAC et RBAC ne permettent pas un contrôle d'accès à niveau fin (tuples ou cellules)
- Oracle propose les VPD (Virtual Private Databases)
  - Ré-écriture de requêtes
  - Politiques de sécurité
  - Fonctions PL/SQL
- Oracle Security Guide 11g, novembre 2012
  http://docs.oracle.com/cd/B28359_01/network.111/b28531.pdf