Droits d'accès

Gestion des droits d'accès

- Modèles de contrôle d'accès
- Commandes `SQL` de contrôle d'accès
- Contrôle de l'usage des bases de données (Chiffrement)

Impératifs de sécurité

- Omniprésence des bases de données
- Information critique
- Nécessité d'autoriser les accès en imposant des restrictions
- Dans certains cas externalisé

Sécurité des bases de données

- Confidentialité
  - Seules les personnes autorisées ont accès aux ressources de la BD
  - Ressources BD: données stockées dans la base ainsi que traitements activables sur ces données
- Intégrité
  - Les ressources de la BD ne doivent pas être corrompues
  - BD: les données stockées et échangées doivent être protégées de toute modification illicite (destruction, altération, substitution)
- Disponibilité
  - L'accès aux ressources de la BD est garanti de façon permanente

Sécurité 1 : Identification/authentification

- Au minimum : login (identification) + password (authentification)
- Assuré par le SGBD ou l'OS
- Authentification forte :
  - Eléments d'authentification distincts parmi :
    - Ce que l'entité connaît (password, ...)
    - Ce que l'entité détient (carte à puce, ...)
    - Ce que l'entité est (biométrie, ...)

Sécurité 2 : Chiffrement des communications

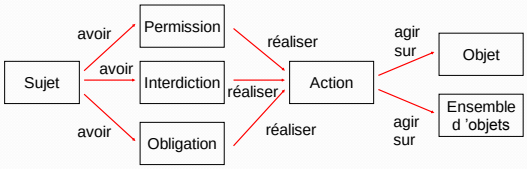
- Communication cryptée entre les utilisateurs et la BD
- Technologie éprouvée (SSL, SSH, ...)
- Assure la confidentialité des messages
- Techniques cryptographiques complémentaires
  - Hash value : intégrité des messages
  - Signature : authentification et non répudiation du message

Sécurité 3 : Contrôle d'accès

- Contrôle d'accès dans les SGBD
  - Autorisations affectées à des utilisateurs, groupes ou rôles
  - Peut porter sur des objets d'une granularité variée : tables, vues, procédures stockées …

Politique de contrôle d'accès

- Ensemble de règles qui précise ce qu'il est autorisé de faire, sur quoi, sur quelles données et sous quelles conditions
- Format des règles :



Modèle discrétionnaire (DAC)

- DAC = Discretionary Access Control
- Principes de DAC
  - Le créateur d'un objet fixe la politique de contrôle d'accès sur cet objet
  - Les autres utilisateurs reçoivent des permissions pour réaliser des actions sur des objets du propriétaire ou d'un administrateur
  - Les autres utilisateurs peuvent avoir l'autorisation de transférer certaines permissions à d'autres utilisateurs

Commandes SQL Grant Privilèges objets

- `GRANT <liste privileges>`
  `ON <table ou vue>`
  `TO <liste utilisateurs>`
  `[ WITH GRANT OPTION ] ;`
- `WITH GRANT OPTION`
  - est optionnel
  - signifie que l 'utilisateur qui obtient le privilège peut ensuite accorder ce privilège à un autre utilisateur
- Ex:
```
GRANT All
  ON Emp
  TO Martin
  WITH GRANT OPTION
```


Privilèges Objets

- Quelques privilèges (permissions) possibles
  - SELECT : permet la consultation de la table
  - INSERT : permet l 'insertion de nouvelles données dans la table
  - UPDATE : permet la mise à jour de n 'importe quelle colonne de la table
  - UPDATE(nom_colonne) : permet la mise à jour d 'une colonne spécifique de la table
  - DELETE : permet de supprimer n 'importe quelle donnée de la table
  - ALTER : Modifier la définition d'un objet
  - EXECUTE : Compiler et exécuter une procédure utilisée dans un programme
  - REFERENCE : référencer une relation dans une contrainte
  - INDEX : Créer un index sur une relation
- Ainsi que les fonctions d'administration: CREATE/ALTER/DROP TABLE et CREATE/DROP USER

Privilèges Systèmes

- Les privilèges donnent le droit de réaliser des opérations systèmes
- Ces privilèges sont classés par catégories d'objets

Exemple:

ANALYZE	AUDIT	CLUSTER
DATABASE	DATABASE	INDEX
MATERIALIZED VIEW	PRIVILEGE	PROCEDURE
PROFILE	PUBLIC SYNONYM	ROLE
ROLLBACK SEGMENT	SESSION	SEQUENCE
SYNONYM	SYSTEM	TABLE
TABLESPACE	TRANSACTION	TRIGGER
TYPE	USER	VIEW

- Exemple de privilèges systèmes de la catégorie TABLE:

CREATE TABLE	CREATE ANY TABLE	ALTER ANY TABLE
DROP ANY TABLE	LOCK ANY TABLE	SELECT ANY TABLE
INSERT ANY TABLE	UPDATE ANY TABLE	DELETE ANY TABLE

Affectation privilège Système

- **Affectation d'un privilège Système**

```
GRANT { system_priv | role }
TO { user | role | PUBLIC }
[ WITH ADMIN OPTION ]
```

system_priv : nom d'un privilège système
role : Nom d'un rôle
user, role ou PUBLIC : droit affecté à un utilisateur, un rôle ou public
With Admin Option : le droit pourra être redistribué par celui qui le reçoit

- L'affectation d'un privilège avec l'option "WITH ADMIN OPTION" suit les règles suivantes :
  - Celui qui reçoit le droit peut le redistribuer
  - Son retrait à un utilisateur qui lui-même l'a affecté à un autre ne peut se faire en cascade
  - Ne peut être affecté à un ROLE
- Dictionnaire : DBA_SYS_PRIVS

Commande SQL Revoke

```
REVOKE [ GRANT OPTION FOR ] <liste privileges>
ON <table ou vue>
FROM <liste utilisateurs>
[option] ;
```

- `[GRANT OPTION FOR]` (pas Oracle, SQL seulement)
  - signifie que seul le droit de transfert est révoqué
- `[option] = RESTRICT` ou `CASCADE` (pas Oracle, SQL seulement)
  - Supposons que A accorde le privilège p à B et B accorde ensuite p à C
  - CASCADE : si A révoque p à B alors C perd aussi le privilège
    - Mode Oracle par défaut
  - RESTRICT : si A révoque p à B alors la révocation échoue
- Ex:

```
REVOKE Delete, Update ON Prescriptions
FROM Martin
CASCADE
```

- Graphe de transmission des droits

Privilèges: dictionnaire

- **Visualisation des privilèges objets**

```
DBA_TAB_PRIVS              DBA_COL_PRIVS        ALL_TAB_PRIVS ALL_COL_PRIVS
USER_TAB_PRIVS             USER_COL_PRIVS       All_TAB_PRIVS_MADE DBA_COL_PRIVS
USER_TAB_PRIVS_MADE        ALL_COL_PRIVS_MADE   USER_TAB_PRIVS_MADE
USER_COL_PRIVS_MADE        ALL_TAB_PRIVS_RECD   ALL_COL_PRIVS_RECD
USER_TAB_PRIVS_RECD        ALL_COL_PRIVS_RECD   TABLE_PRIVILEGES COLUMN_PRIVILEGES
…
```

- **Principales Colonnes de vues ci-dessus**

```
GRANTEE : utilisateur ayant reçu le privilège
OWNER : propriétaire de la table
TABLE_NAME : nom de la table
COLUMN_NAME : Nom de la colonne concerné
GRANTOR : Utilisateur ayant affecté le privilège
PRIVILEGE : privilège affecté
GRANT : privilège reçu.
```


Privilèges: consultation du dictionnaire

- **Visualisation des privilèges objets**

Visualisation de tous les droits sur les objets de la base

```
SELECT * FROM sys.dba_tab_privs
WHERE table_name = 'BONUS' OR
table_name = 'EMP';

Affichage =>
```

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE
DUPONT	SCOTT	BONUS	SCOTT	ALTER
DUPONT	SCOTT	BONUS	SCOTT	DELETE
DUPONT	SCOTT	BONUS	SCOTT	INDEX
DUPONT	SCOTT	BONUS	SCOTT	INSERT
DUPONT	SCOTT	BONUS	SCOTT	SELECT
DUPONT	SCOTT	BONUS	SCOTT	UPDATE
DUPONT	SCOTT	BONUS	SCOTT	REFERENCES

Tous les droits sur toutes les colonnes des tables dans la base

```
SELECT * FROM sys.dba_col_privs ;
```

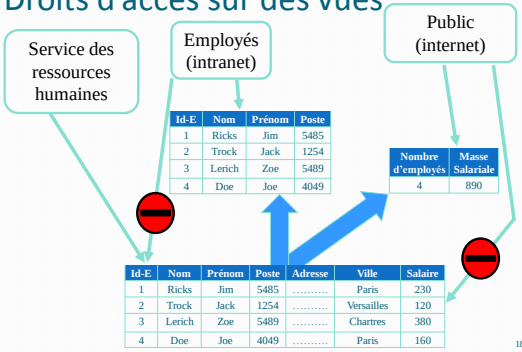

Sécurité: utilisation des vues

Exemple:

```
CREATE OR REPLACE VIEW empInge AS
       SELECT * from emp
       WHERE job= 'Ingenieur'
       WITH CHECK OPTION  CONSTRAINT Check_inge;
```

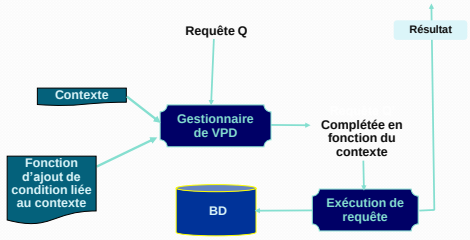
- Choix des colonnes résultats (SELECT)
  ⇒Transmettre le droit de consulter une vue permet de restreindre les colonnes résultats
- Choix des n-uplets résultats (SELECT)
  ⇒Transmettre le droit de consulter une vue permet d'imposer des contraintes que devront respecter les n-uplets résultats
- Choix des n-uplets modifiées (Vue WITH CHECK OPTION, INSERT, DELETE)
  ⇒Transmettre le droit de modifier/insérer sur une vue permet d'imposer des contraintes que devront respecter les n-uplets à modifier

Droits d'accès sur des vues



Base de données privée virtuelle (VPD)

Principe (Oracle) : Rajouter dynamiquement des conditions aux requêtes utilisateur en fonction d'un contexte quelconque (lié à cet utilisateur et/ou à l'application)



VPD : Contexte d'application

- Oracle a prévu un contexte par défaut
  - USERENV : contient des informations système relatives à la session courante
  - Exemple : CURRENT_USER, HOST, ISDBA
- Possibilité de créer un nouveau contexte et d'y associer des attributs
  - Exemple :
    - `create context CTX_SEC_MEDICALE using SCHEMA_MED.SEC_MEDICALE`
    - `CTX_SEC_MEDICALE` sera un contexte associé au package PL/SQL nommé `SEC_MEDICALE` et stocké dans le schéma `SCHEMA_MED`

VPD : Définition des règles de sécurité

- Les règles de sécurité sont écrites en PL/SQL

```
create package body SEC_MEDICALE as
  function DOSSIER_SEC return varchar2 is
    MY_PREDICATE varchar2(2000) ;
    BEGIN
    MY_PREDICATE := 'id_patient in
        (SELECT id_patient
         FROM dossier_medical
         WHERE medecin_traitant =
           sys_context("USERENV", "CURRENT_USER"))' ;
    return MY_PREDICATE ;
  END DOSSIER_SEC ;
END SEC_MEDICALE ;
```

- Puis associées à un objet

```
DBMS_RLS.add_policy    // RLS = Row Level Security
     (object_schema      => 'SCHEMA_MED',
      object_name        => 'dossier_medical',
      policy_name        => 'mesdossiers',
      policy_function    => ' DOSSIER_SEC');
      END;
```


Exemple de transformation de requêtes

- Supposons que le médecin Martin formule la requête suivante :

```
SELECT *
FROM dossier_medical
WHERE id_patient = 'Scott'
```

- L'application de la règle *mesdossiers* va automatiquement transformer cette requête en la requête suivante :

```
SELECT *
FROM dossier_medical
WHERE id_patient = 'Scott'
AND   id_patient in       (SELECT id_patient
                           FROM dossier_medical
                           WHERE medecin_traitant =
  'Martin' ) ;
```

- Martin ne pourra ainsi accéder qu'aux dossiers médicaux de ses patients

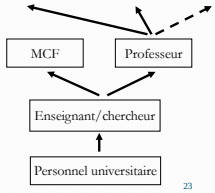
RBAC : Role-Based Access Control

- Rôle = ensemble de privilèges associés à une fonction
- Les utilisateurs sont habilités à jouer certains rôles
- Les rôles peuvent être organisés en hiérarchie
- Factorise la gestion des privilèges

- R1 rôle senior de R2

  si chaque fois qu'un utilisateur joue le rôle R1, cet utilisateur joue aussi le rôle R2

- Permet d'exprimer
  - Spécialisation/Généralisation
  - Relation hiérarchique entre utilisateurs



RBAC : Gestion des rôles dans SQL

- Création des rôles
  - `CREATE ROLE <nom_role> ;`
    - Création d'un nouveau rôle nom_role
  - `DROP ROLE <nom_role> ;`
    - Suppression du rôle nom_role
  - `SET ROLE <liste_roles> ;`
    - Permet à un utilisateur d 'activer un ensemble de rôles pendant la durée d'une session SQL
- Affectation des privilèges aux rôles

```
GRANT <liste privileges>
ON <table ou vue>
TO <liste roles>
[ WITH GRANT OPTION ] ;
```

- Affectation des utilisateurs aux rôles

```
GRANT <liste roles>
TO <liste utilisateurs>
```

- Rôle junior et rôle senior

  `GRANT <role1> TO <role2>`

  Le rôle role2 reçoit tous les privilèges du rôle role1

Exemple d'utilisation

- Le médecin a accès aux dossiers de ses patients
- Définition d'une vue

```
Create view dossier_patient_du_medecin as
Select * From dossier_patient
Where dossier_patient.medecin_traitant=CURRENT_USER;
```

- Création du rôle médecin

```
Create role medecin
```

- Affectation des droits au rôle médecin

```
Grant all on dossier_patient_du_medecin to medecin
```

- Affectation du rôle aux médecins

```
Grant medecin to Martin, Muller;
```


Profiles

- Un profile est un concept Oracle qui permet à l'administrateur d'une base de contrôler la consommation des ressources systèmes et de poser des contraintes sur les mots de passe.
- Il existe un profile par défaut appelé DEFAULT. Il est par défaut affecté à un utilisateur lors de sa création.
- Les limites du profile DEFAULT sont positionnées à UNLIMITED.
- Le profile DEFAULT ne peut être supprimé. Les limites de ce profile peuvent par contre être modifiées.

Profiles

Création d'un profile
Privilège requis
CREATE PROFILE

Syntaxe partie limite des ressources

```
CREATE PROFILE profile LIMIT
[ SESSIONS_PER_USER { integer | UNLIMITED | DEFAULT} ]
[ CPU_PER_SESSION { integer | UNLIMITED | DEFAULT } ]
[ CPU_PER_CALL { integer | UNLIMITED | DEFAULT } ]
[ CONNECT_TIME { integer | UNLIMITED | DEFAULT } ]
[ IDLE_TIME { integer | UNLIMITED | DEFAULT } ]
[LOGICAL_READS_PER_SESSION {integer | UNLIMITED|DEFAULT}]
[LOGICAL_READS_PER_CALL {integer | UNLIMITED|DEFAULT}]
[ COMPOSITE_LIMIT { integer | UNLIMITED | DEFAULT } ]
[PRIVATE_SGA {integer [K | M] | UNLIMITED | DEFAULT}]
```

Mots clés et paramètres

Session_per_user :	nombre maximum de sessions par utilisateur
Logical_read_per_session :	nbre de blocs de données à lire pour une session
cpu_per_session :	temps CPU max par session en % de secondes
cpu_per_call :	temps CPU pour un appel (en cas de parse, execute ou fetch) en % de secondes
connect_time :	temps écoulé maximum (en minutes)
idle_time :	temps maximum d'inactivité.
private_sga :	taille privée de la SGA allouée à un utilisateur
unlimited :	limite de la ressource illimitée
default :	prend la limite par défaut de la ressource

Utilisateur Oracle

- La notion d'utilisateur est fondamentale pour accéder aux données d'une base Oracle.
- A chaque utilisateur est associé un schéma
- Les objets appartenant à un schéma sont : tables, index, vues, séquences, synonymes, clusters, fonction, procédures et package, ...

Utilisateur Oracle

Création d'un utilisateur
Lors de la création d'un utilisateur, il est possible de lui affecter : un mot de passe, un tablespace par défaut, un tablespace temporaire, un profile (explicite ou implicite), des quotas sur les tablespaces.

Syntaxe

```
CREATE USER user
IDENTIFIED { BY password | EXTERNALLY| GLOBALLY
AS 'nom_externe' }
[ DEFAULT TABLESPACE tablespace ]
[ TEMPORARY TABLESPACE tablespace ]
[ QUOTA { integer [ K | M ] | UNLIMITED } ON tablespace ] ...
[ PROFILE profile ]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
```

Mots clés et paramètres
user : nom de l'utilisateur à créer
password : mot de passe
externally : utilisateur authentifié par l'OS
tablespace : nom du tablespace
profile : nom du profile
globally as : accès autorisé l'annuaire LDAP

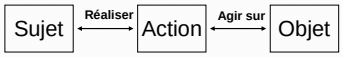
Un rôle par défaut peut être affecté à un utilisateur par la commande ALTER USER

MAC : Mandatory Access Control

- Objectif : lutter contre les chevaux de Troie
  - transmission illégale d'informations via un programme malveillant
- Politique de sécurité multi-niveaux (simplifiée)
  - Niveaux de sécurité hiérarchiques
    - **Unclassified => Confidential =>Secret =>Top Secret**
  - Le niveau de sécurité d'un utilisateur = niveau d'accréditation
  - Le niveau de sécurité d'un objet = niveau de classification
- Principe du contrôle
  - No read up : read permis si accréditation ≥ classification
  - No write down : write permis si accréditation ≤ classification (permet d'éviter attaque type cheval de trois: copie d'informations dans une zone d'un niveau de sécurité inférieur)

Synthèse sur les modèles de contrôle d'accès

- Principe fondateur



- DAC
  - Permet de structurer les Objets
- RBAC
  - Permet de structurer les Utilisateurs
- MAC
  - Lutte contre les programmes malveillants
  - Mais offre peu de souplesse dans la définition des politiques
- Ces modèles ne s'appliquent que pour les utilisateurs « entrant » dans le SGBD par des moyens classiques

Sécurité 4 : Chiffrement de la BD

- Principe : chiffrer l'empreinte disque de la BD pour résister aux attaques sur les fichiers
- Chiffrement sur disque ? En mémoire principale ?

Surveiller les accès à la base:audit

- L'objectif de l'audit est de contrôler les accès mal intentionnés ou non autorisés à la base
- Bien cibler l'audit pour en limiter le volume
- protéger la table d'audit (sys.aud$) en limitant l'affectation du privilège DROP ANY TABLE
- Types d'audit
  - par utilisateur : auditer l'activité d'un ou plusieurs utilisateurs
  - par session (en cas de succès ou d'insuccès) : Pour N activations d'une action dans une session, conserver une seule trace
  - par accès (en cas de succès ou d'insuccès) : Pour N activations d'une action dans une session, conserver N trace
- Types de surveillance
  - Audit sur les ordres SQL et les privilèges systèmes
  - Audit sur les objets.

Audit

- **Audit des ordres sql et des privilèges systèmes**
  Privilège requis : AUDIT SYSTEM
- **Syntaxe**

```
AUDIT { {statement_opt | ALL} |
{system_priv | ALL PRIVILEGES} }
[{ {statement_opt | ALL} |
{system_priv | ALL PRIVILEGES} }]
[ BY user [, user ] ...
[ BY { SESSION | ACCESS } ]
[ WHENEVER [ NOT ] SUCCESSFUL ]
```

- **Mots clés et paramètres**
  - statement_opt : option des ordres sql à auditer
  - system_priv : privilège système à auditer
  - user : audit des ordres sql et/ou privilèges pour un user
  - BY SESSION : audit des ordres sql et/ou privilèges par session
  - BY ACCESS : audit des ordres sql et/ou privilèges par accès
  - SUCCESSFUL : audit des ordres sql et/ou privilèges si succès
  - ALL : audit de tous les ordres SQL
  - ALL PRIVILEGES : audit de tous les privilèges

Audit

- Audit des ordres SQL et des privilèges systèmes
- Exemples

auditer les connexions et déconnexions des utilisateurs Martin et Scott (en cas de succès ou d'insuccès).

```
AUDIT SESSION BY Martin, Scott;
```

auditer les ordres select, insert, delete pour toutes les tables et le privilège execute any procedure par accès et ceci en cas d'échec.

```
AUDIT     select table, insert table,
          delete table, execute any procedure
BY ACCESS WHENEVER NOT SUCCESSFUL ;

AUDIT table, execute any procedure
BY ACCESS WHENEVER NOT SUCCESSFUL ;
```

auditer l'ordre SQL select sur la table Martin.emp en cas d'échec d'exécution de cet ordre.

```
AUDIT select ON Martin.emp WHENEVER NOT SUCCESSFUL ;
```

auditer l'insertion, la modification et la sélection sur la table Scott.dept par accès.

```
AUDIT insert, update, select ON Scott.dept BY ACCESS ;
```


Audit

- Exploitation de l'audit
  la table de base AUD$ contient l'ensemble des informations sur les audits
- Les vues d'audit :

USER_AUDIT_CONNECT	USER_AUDIT_RESOURCE
DBA_AUDIT_DBA	DBA_PRIV_AUDIT_OPTS
DBA_TAB_AUDIT_OPTS	ALL_DEF_AUDIT_OPTS
DBA_STMT_AUDIT_OPTS	USER_OBJ_AUDIT_OPTS
DBA_OBJ_AUDIT_OPTS	USER_AUDIT_TRAIL, DBA_AUDIT_TRAIL
USER_AUDIT_STATEMENT	DBA_AUDIT_STATEMENT
USER_AUDIT_OBJECT	DBA_AUDIT_OBJECT
DBA_AUDIT_EXISTS	USER_AUDIT_SESSION
DBA_AUDIT_SESSION	

- Ex:

```
SELECT * FROM dba_obj_audit_opts
WHERE owner = 'SCOTT' AND object_name like 'EMP%';
```

Métiers autour des BD

- Administrateur
- Responsable de la sécurité
- Administrateur réseaux
- Développeurs d'application
- Administrateurs d'application
- Utilisateurs : modifier les données, créer des rapports

37

Rôle du DBA

- Planifier et créer des bases de données
- Gérer l'espace mémoire et implanter les schémas des données
- Assurer la sécurité, l'intégrité et la pérennité des données
- Effectuer des réglages pour optimiser les performances

38