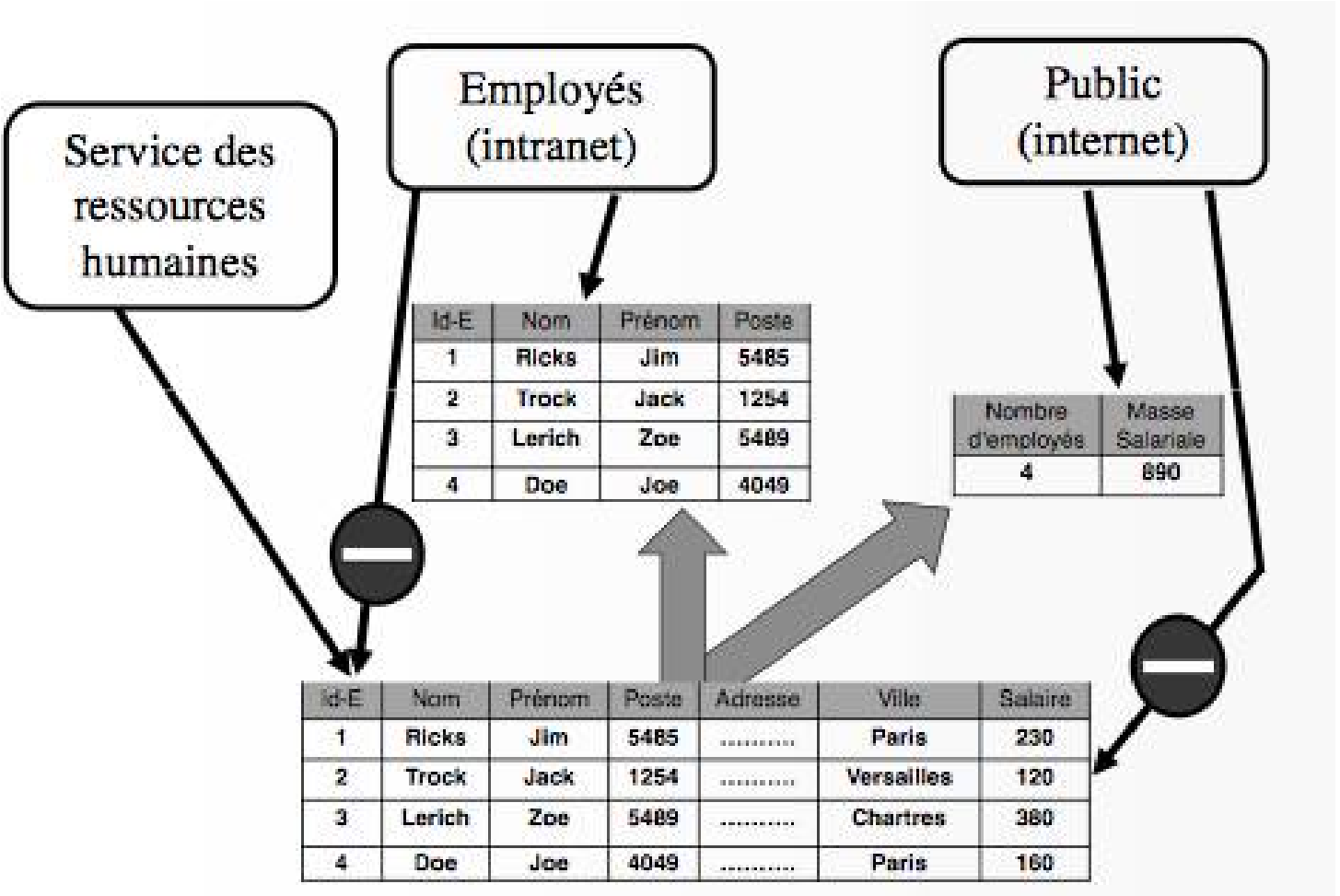
Retour sur le TP

Confidentialité via les vues



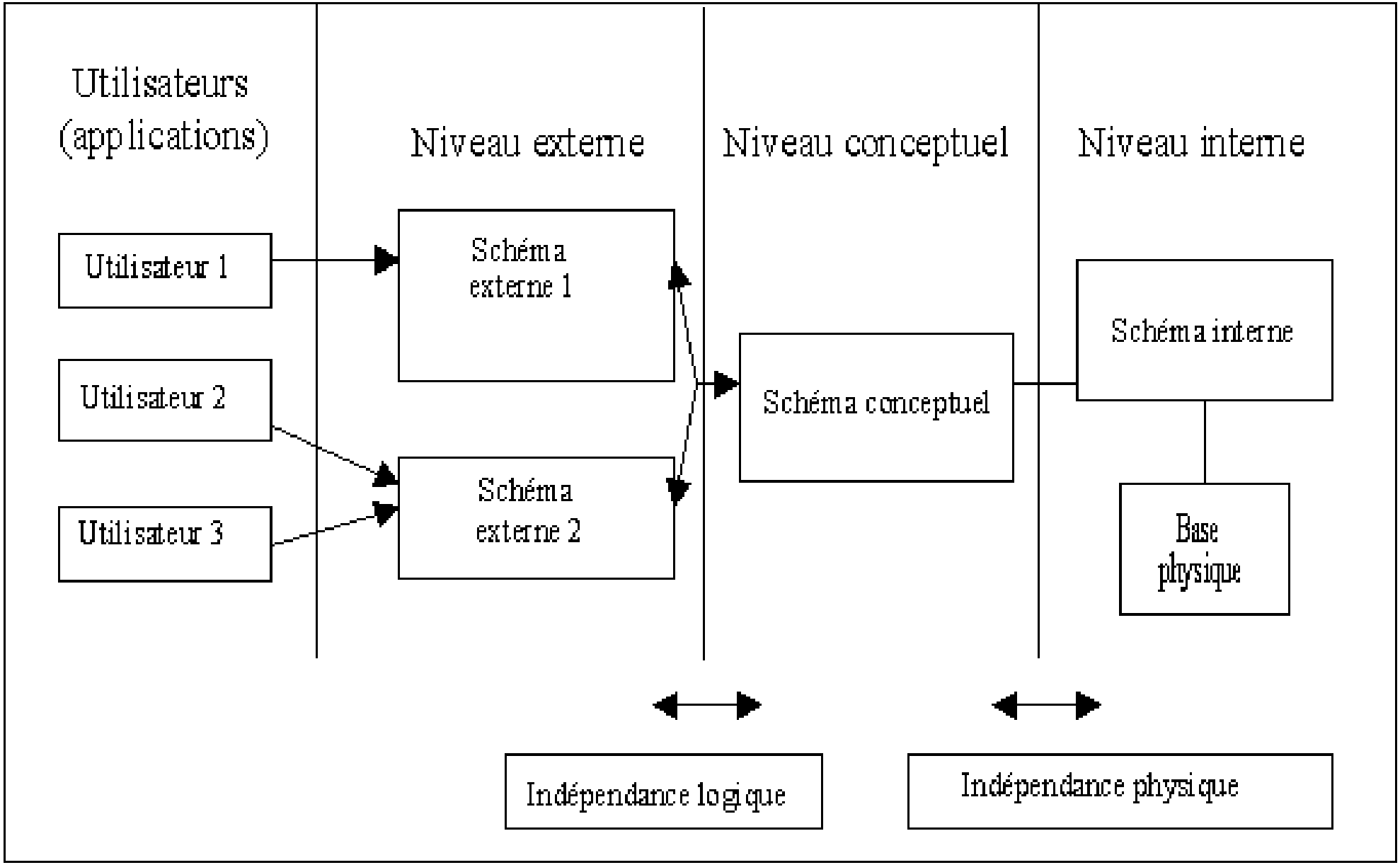
http://www-smis.inria.fr/~pucheral/

Les vues

Indépendance logique :

- Une application peut ignorer les besoins des autres
- La base de données peut évoluer sans avoir à réécrire les applications
- Limiter le partage des données (confidentialité)

Architecture à 3 niveaux



Architecture à 3 niveaux

- Proposition du groupe ANSI/X3/SPARC
- Abstraction des données stockées sur disques pour simplifier la vision des utilisateurs
- Niveau interne : structure de stockage des données (fichiers, chemins d'accès, ...)
- Niveau conceptuel : structure sémantique des données (types d'objets et liens entre ces objets), sans souci de l'implantation (ex. schéma de la base de données d'une entreprise)
- Niveau externe : description d'une partie des données utiles pour un groupe d'utilisateurs (ex. données utiles à la comptabilité)

Schéma conceptuel

Vision globale de la base de données

- décrit les objets à gérer (personne, service)
  - leur structure : propriétés, domaines (ou types)
  - les contraintes d'intégrité

Schémas externes

Vision propre à chaque application

- dérivé du schéma conceptuel
  - vue partielle
  - assemblage différent des données de la base
- immunise les applications contre la plupart des modifications du schéma conceptuel (indépendance logique)
- permet d'assurer une sécurité/confidentialité des données
- en général, il y en a plusieurs

Schéma interne

Comment les données sont stockées

- fichiers
- chemins d'accès (index), ...
- divers paramètres pour rendre l'accès aux données plus efficace

Architecture à 3 niveaux

Permet d'assurer indépendance physique et logique

Indépendance logique : programmes d'application insensibles aux modifications apportées au schéma conceptuel qui ne les concernent pas

Exemple : l'ajout de la propriété date de naissance à l'objet personne n'entraîne pas de modifications dans le code de l'application de comptabilité

Architecture à 3 niveaux

Indépendance physique : applications indépendantes vis à vis de la représentation interne des données

Exemple : l'ajout ou la suppression d'un index à un objet n'affecte pas les programmes utilisant cet objet

=> Indépendance physique et logique permet d'éviter une maintenance coûteuse des applications en cas de modifications du schéma conceptuel ou interne (mode de stockage)

Confidentialité via les vues/ droits

Pourquoi utiliser les vues et pas seulement les droits ?

Confidentialité via les vues/ droits

- Gestion plus fine (exemple du TP : un employé peut connaître ses informations uniquement)

Autre exemple (rh_analystes est un rôle pour les RH s'occupant des analystes uniquement) :

- Sans les vues

```
GRANT select(empno, ename, sal, deptno)
ON emp TO rh_analystes ;
```

Les RH des analystes voient aussi les informations sur les non analystes

- Avec les vues

```
CREATE VIEW infos_rh_analystes
AS (SELECT empno, ename, sal, deptno
    FROM emp WHERE lower(job)='analyst')
GRANT select ON infos_rh_analystes TO rh_analystes
```

La gestion se fait au niveau du contenu !

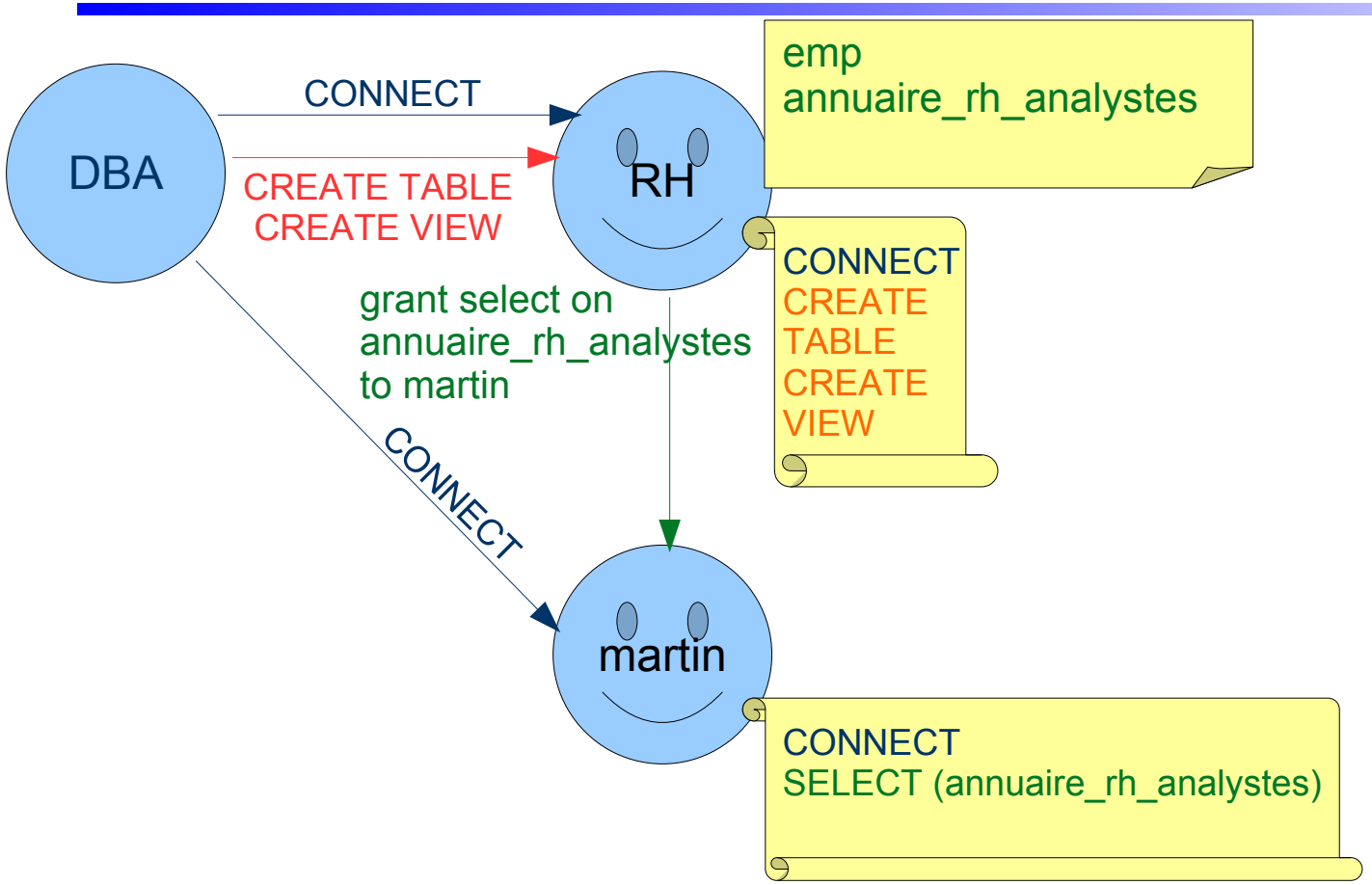
Confidentialité via les vues/ droits

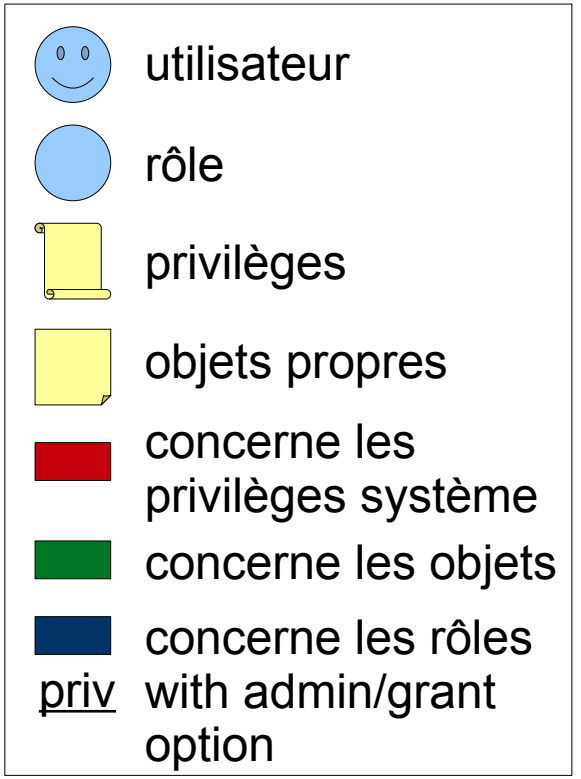
- Possibilité de donner des droits sur des informations multi-tables

```
CREATE VIEW infos_rh_analystes
AS (SELECT e.empno, e.ename, e.sal, d.dname
    FROM emp e JOIN dept d ON e.deptno=d.deptno
    WHERE lower(e.job)='analyst')


GRANT select ON infos_rh_analystes
TO rh_analystes ;
```

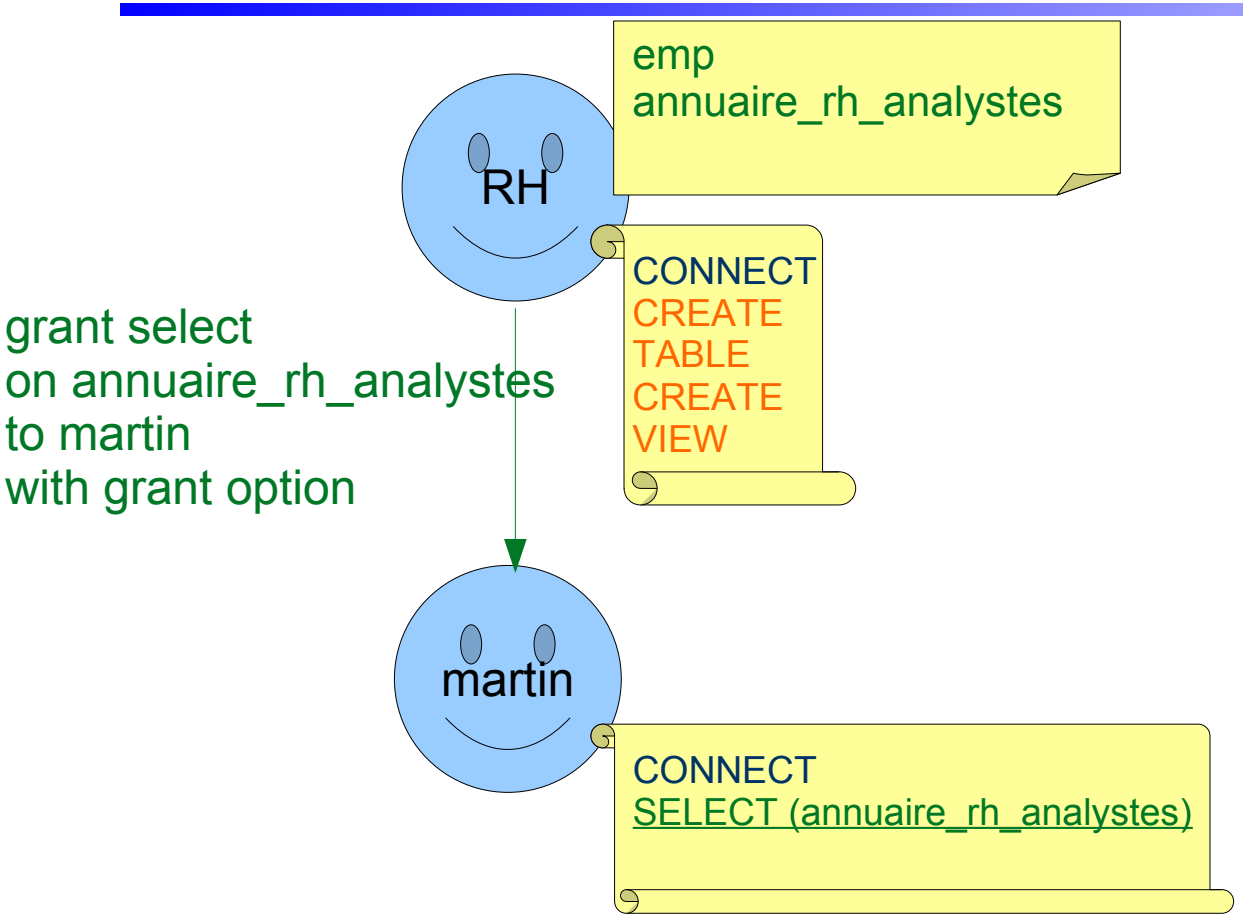
Les droits

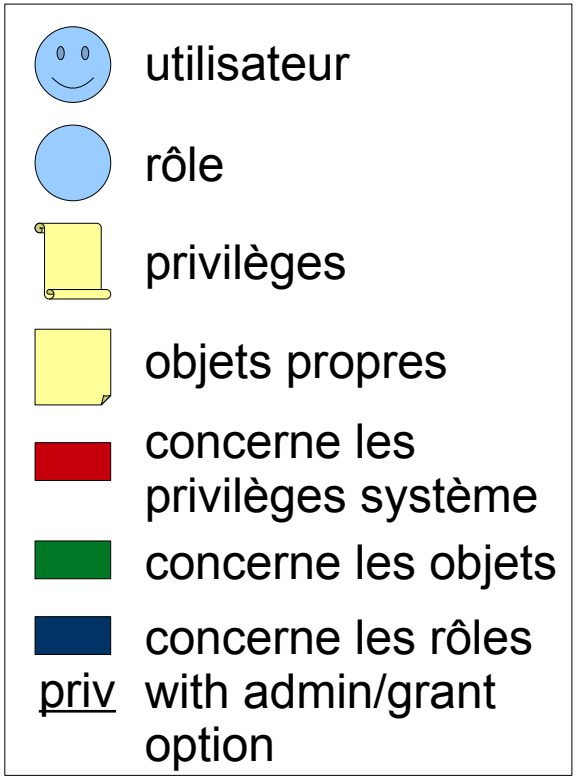




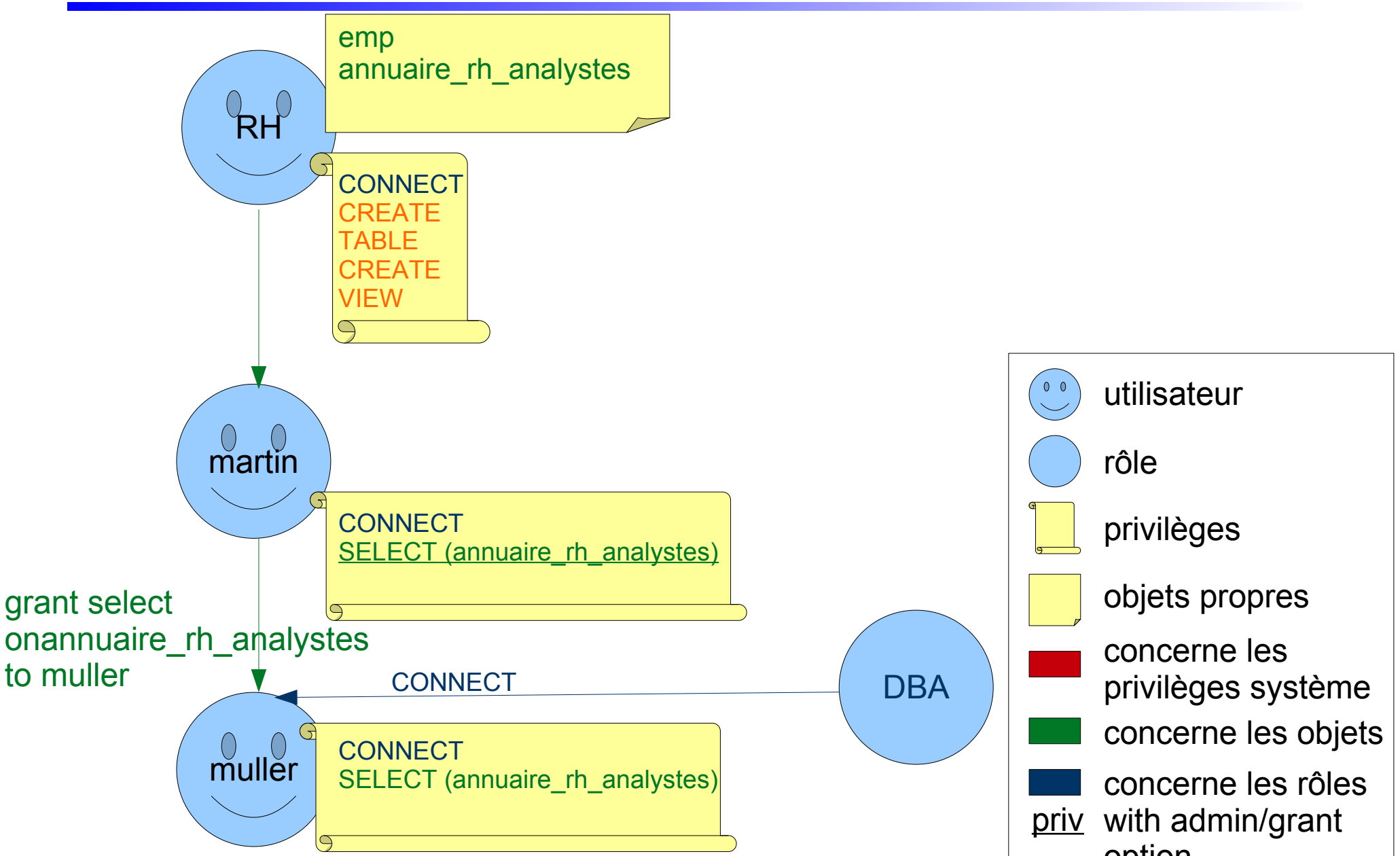
CONNECT permet de se connecter à la base, on peut ensuite faire des SELECT (compte DUT)

Les droits

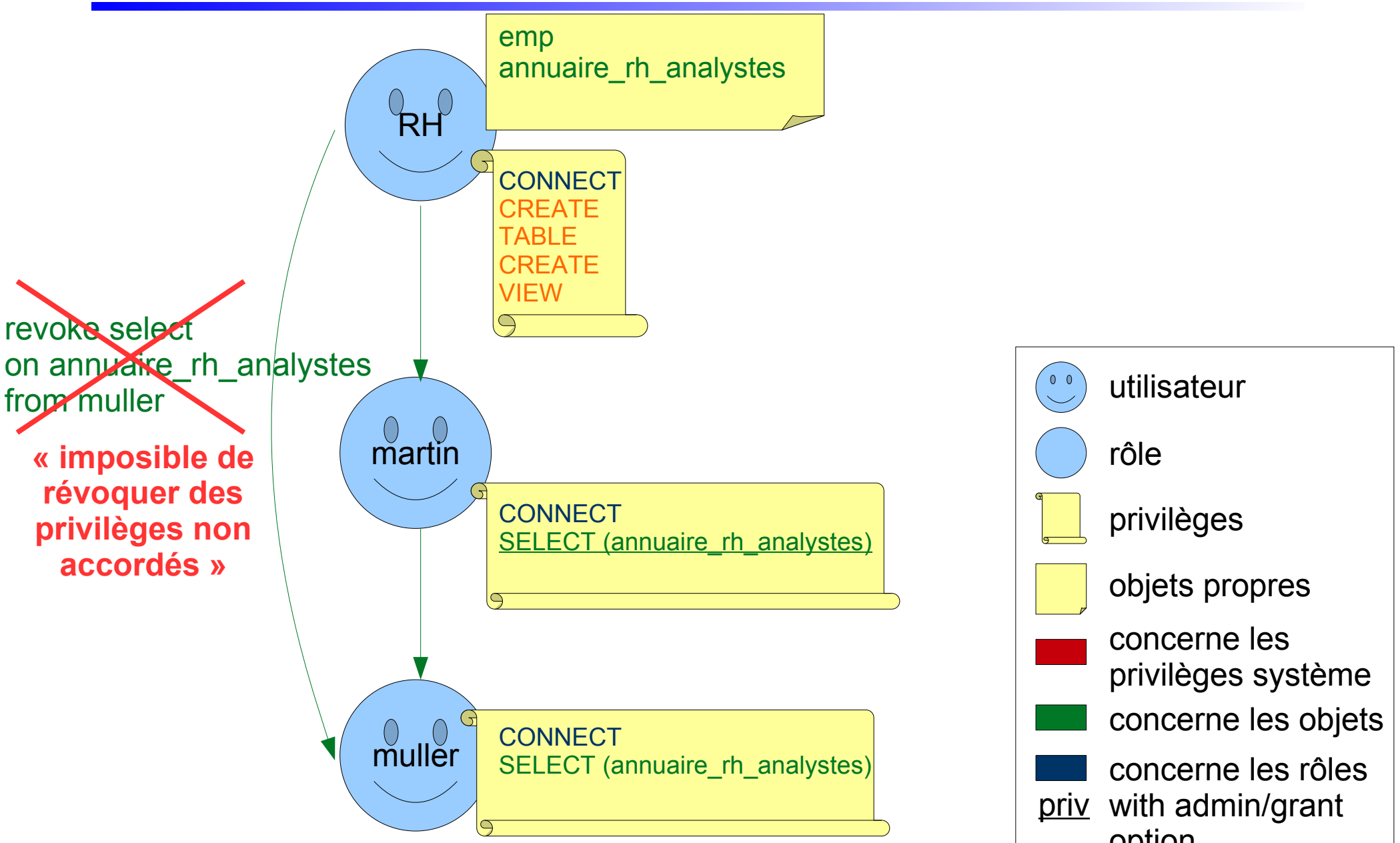

emp
annuaire_rh_analystes
RH
CONNECT
CREATE TABLE
CREATE VIEW
grant select
on annuaire_rh_analystes
to martin
with grant option
martin
CONNECT
SELECT (annuaire_rh_analystes)


utilisateur
rôle
privilèges
objets propres
concerne les privilèges système
concerne les objets
concerne les rôles
priv
with admin/grant option

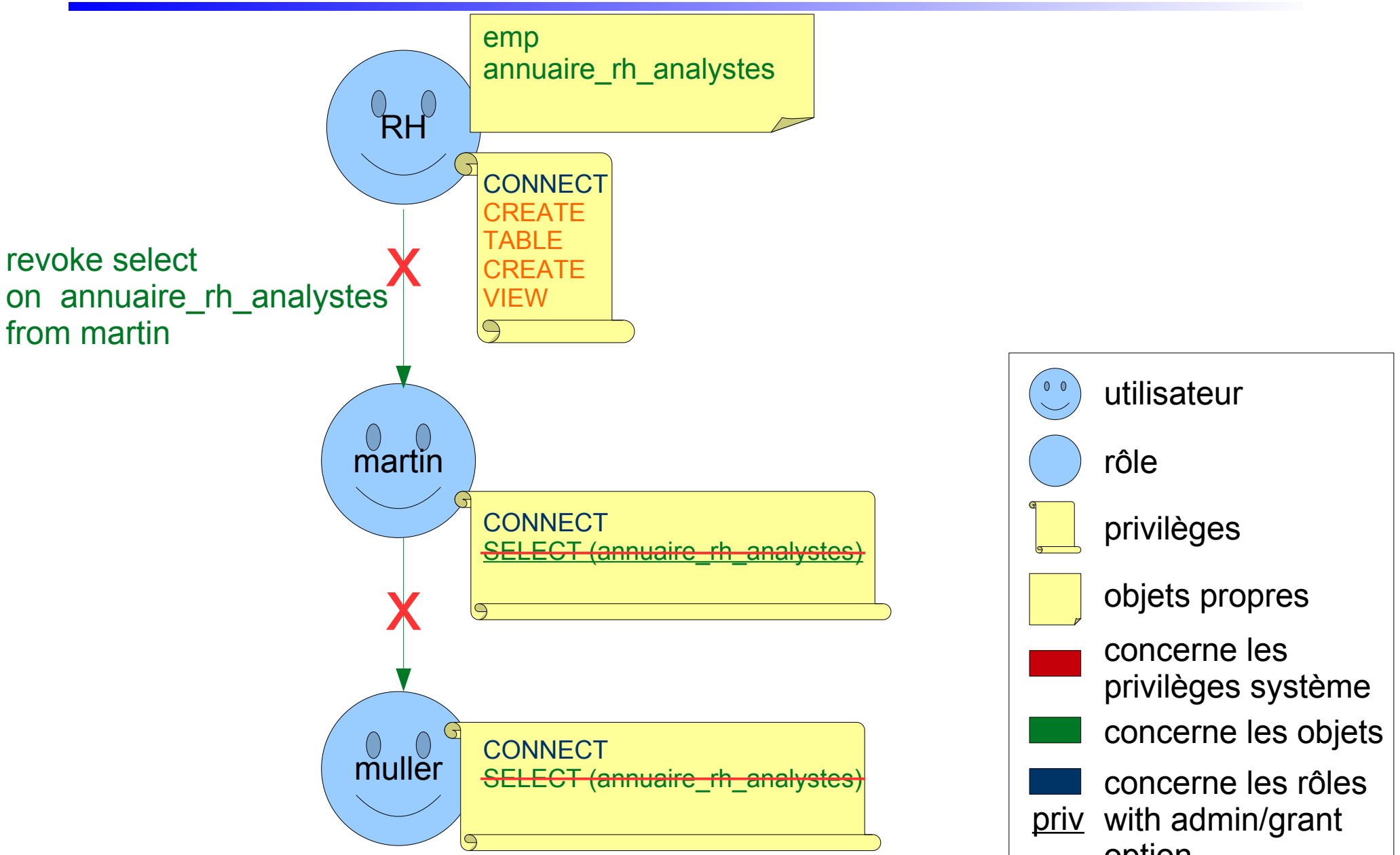
Les droits


emp
annuaire_rh_analystes
RH
CONNECT
CREATE TABLE
CREATE VIEW
martin
CONNECT
SELECT (annuaire_rh_analystes)
grant select
onannuaire_rh_analystes
to muller
CONNECT
DBA
muller
CONNECT
SELECT (annuaire_rh_analystes)
utilisateur
rôle
privilèges
objets propres
concerne les privilèges système
concerne les objets
concerne les rôles
priv
with admin/grant option

Les droits


emp
annuaire_rh_analystes
RH
CONNECT
CREATE TABLE
CREATE VIEW
revoke select
on annuaire_rh_analystes
from muller
« impossible de révoquer des privilèges non accordés »
martin
CONNECT
SELECT (annuaire_rh_analystes)
muller
CONNECT
SELECT (annuaire_rh_analystes)
utilisateur
rôle
privilèges
objets propres
concerne les privilèges système
concerne les objets
concerne les rôles
priv with admin/grant option

Les droits


emp
annuaire_rh_analystes
RH
CONNECT
CREATE TABLE
CREATE VIEW
revoke select
on annuaire_rh_analystes
from martin
martin
CONNECT
SELECT (annuaire_rh_analystes)
muller
CONNECT
SELECT (annuaire_rh_analystes)
utilisateur
rôle
privilèges
objets propres
concerne les privilèges système
concerne les objets
concerne les rôles
priv with admin/grant option