EISTI
Ecole Internationale des Sciences du Traitement de l'Information

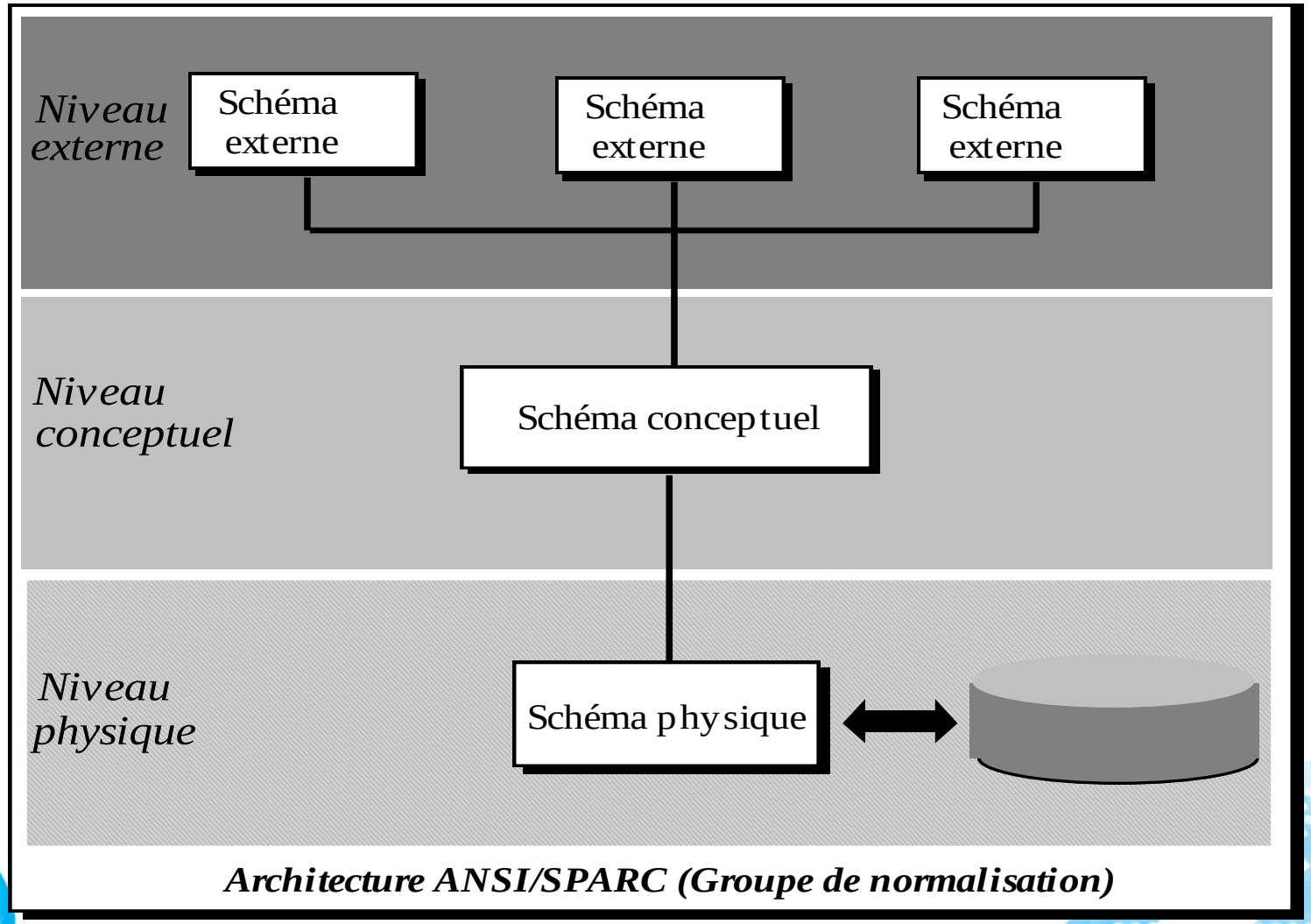
Bases de données avancées

SPARC : Niveau externe

2012/2013

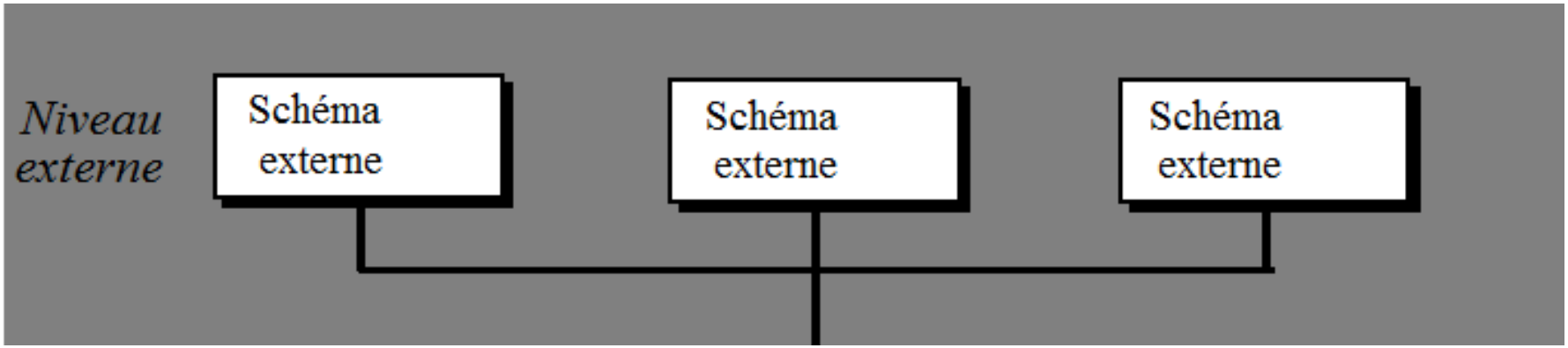
- Architecture SPARC
- Utilisateurs et rôles
- Privilèges systèmes
- Privilèges objets
- Vues

Architecture SPARC



Architecture ANSI/SPARC (Groupe de normalisation)

Architecture SPARC



- Les utilisateurs
- Les rôles
- Les vues logiques et matérialisées

Le concept d'utilisateur

- Quand on crée un utilisateur sous oracle, on précise son nom, son mot de passe et son espace logique de travail

  CREATE USER nomUser IDENTIFIED BY motDePasse DEFAULT TABLESPACE users;

- A sa création, un utilisateur ne peut rien faire, il n'a aucun rôle.
- A la création d'une base de données, deux rôles sont définis : SYSTEM et SYS

Le concept de rôle

- Un rôle est un ensemble nommé de privilèges. Un privilège est la possibilité d'exécuter une action sur la base de données
- Un rôle peut être attribué à des utilisateurs ou à d'autres rôles.

Le concept de rôle

- A sa création un rôle est vide.
- On attribue des privilèges ou des rôles à un rôle avec l'instruction grant

GRANT privilege | role TO role;

- On retire des privilèges ou des rôles d'un rôle avec l'instruction revoke

REVOKE privilege | role FROM role;

Rôle d'application

- Un rôle peut être associé à un package. On dit alors que c'est une rôle d'application.
- Un tel rôle est automatiquement activé lors de l'exécution d'un élément du package
- C'est l'unique façon de l'activer.

Rôle utilisateur

- Les autres rôles sont appelés rôles utilisateurs.
- Un tel rôle peut être protégé par un mot de passe

  CREATE ROLE nomRole [NOT IDENTIFIED|IDENTIFIED {by mot_de_passe}]
- Un rôle protégé doit être activé par l'ordre SET

  SET ROLE nomRole IDENTIFIED BY mot_de_passe;

Rôles prédéfinis

Oracle a prédéfini des rôles : les plus connus

Rôle	Privilèges système
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, UNLIMITED TABLESPACE
DBA	Tous les privilèges
EXP_FULL_DATABASE	SELECT ANY TABLE, BACKUP ANY TABLE
IMP_FULL_DATABASE	BECOME USER

Exemples : Utilisateurs et rôles

```
CREATE USER Udev IDENTIFIED BY motdepasse;
CREATE ROLE Rdev ;
GRANT CONNECT,RESOURCE to Rdev ;
GRANT Rdev to Udev ;
```

CONNECT et RESOURCE sont des rôles prédéfinis dans Oracle.

Les privilèges Systèmes

Un privilège Système permet de créer des objets Oracle

LES PRIVILEGES SYSTEME	ALTER	CREATE	DROP
ANY CLUSTER	X	X	X
CLUSTER		X	
ANY INDEX	X	X	X
PROCEDURE		X	
ANY PROCEDURE	X	X	X
ROLE		X	
ANY ROLE	X		X
SEQUENCE		X	
ANY SEQUENCE	X	X	X
SNAPSHOT		X	
ANY SNAPSHOT	X	X	X
TABLE		X	
ANY TABLE	X	X	X
TRIGGER		X	
ANY TRIGGER	X	X	X

Les privilèges Systèmes

LES PRIVILEGES SYSTEME	ALTER	CREATE	DROP
DATABASE	X		
PROFILE	X	X	X
RESOURCE COST	X		
ROLLBACK SEGMENT	X	X	X
SESSION	X	X	
SYSTEM	X		
TABLESPACE	X	X	X
USER		X	X
VIEW		X	
ANY VIEW		X	X
ANY SYNONYM		X	X
SYNONYM		X	
DATABASE LINK		X	X
PUBLIC DATABASE LINK		X	
PUBLIC SYNONYM		X	X

Les privilèges Objets

Droit d'exécuter une action particulière sur :
une table, une vue, une séquence, une procédure, une fonction, un package ou une vue matérialisée

Les privilèges objets	Table	Vue	Séquence	Procédure Fonction Package	materialized view
ALTER	X		X		
DELETE	X	X			
EXECUTE				X	
INDEX	X				
INSERT	X	X			
REFERENCES	X				
SELECT	X	X	X		X
UPDATE	X	X			

Les privilèges Objets : Exemples

Droits sur une table pour un utilisateur

GRANT SELECT, INSERT, UPDATE, DELETE ON SUPPLIERS TO SMITHJ;

Droits sur toutes les tables pour un utilisateur

GRANT SELECT ANY TABLE TO FRED;

Droits sur une clé étrangère pour un utilisateur

GRANT REFERENCES (CLI_ID) ON TABLE T_CLIENT TO DUMONT;

Droit d'utilisation d'une séquence pour un rôle

GRANT ALTER ON SEQ_ID_CLIENT TO RESPONSABLE_CLIENT;

Droit d'exécution d'une procédure pour un utilisateur

GRANT EXECUTE ON PROC_MAJ_PRIX TO PIERROT;

Droit de sélection sur une vue matérialisée pour un rôle

GRANT SELECT ON VM_PRODUITS TO RESPONSABLE_CLIENT;

Les vues

Pour limiter l'accès à des données, on dispose, en plus des rôles, de vues :

- des vues logiques : c'est une requête SQL qui est ré-exécutée à chaque appel. Il n'y a donc aucun stockage physique
- des vues matérialisées : c'est une requête SQL dont le contenu est dupliqué à la création et rafraîchit régulièrement. Il y a donc un stockage physique.

Pourquoi des vues ?

• Dans les deux cas, elles permettent de filtrer des données. C'est donc un outil essentiel dans les schémas externes.

Exemple

- Table : create table Client(nom …, prenom …, age …)
- Vue : create view Vue_Client select nom,prenom from Client

Cette vue permet de masquer l'accès à l'âge.

• Dans le cas de la vue matérialisée, il s'agit de gain de performances pour les requêtes compliquées comportant des jointures

• Une base de données bien construite est normalisée.

• Créer des vues matérialisées consiste en général à dénormaliser pour des soucis de performances

Les vues matérialisées (VM)

- La création la plus basique d'une vue matérialisée

CREATE MATERIALIZED VIEW MVue_Client AS SELECT * FROM Client

- Il faut avoir les privilèges
  - système : CREATE MATERIALIZED
  - objet : GRANT SELECT ON sur les tables de la requête

- Par défaut, la table matérialisée est créée avec le contenu de la sélection associée.

- Si on veut remplir la vue matérialisée plus tard

CREATE MATERIALIZED VIEW MV_Client BUILD DEFERRED
AS SELECT * FROM Client

- Dans ce qui suit, on pourra avoir besoin que la vue contienne la clé primaire. Il est donc fortement conseillé de satisfaire cette condition.

VM: Types de Raffraichissement

- On peut rafraichir de trois types :
  - à la demande en exécutant la procédure dbms_mview.refresh :

    CREATE MATERIALIZED VIEW MV_Client REFRESH ON DEMAND
    AS SELECT * FROM Client;

    puis quand on veut : EXECUTE DBMS_MVIEW.REFRESH('MV_Client');

    C'est la méthode par défaut.

  - pour toute modification validée (commit) d'une des tables de la requête

    CREATE MATERIALIZED VIEW MV_Client REFRESH ON COMMIT
    AS SELECT * FROM Client;

  - à intervalle régulier

    CREATE MATERIALIZED VIEW MV_Client REFRESH START WITH Date_Depart NEXT temps_en_jour AS SELECT * FROM Client;. temps_en_jour peut être fractionnaire.

VM: Méthodes de rafraîchissement

- Il existe trois méthodes de rafraîchissement :
  - FAST : Oracle ne rafraîchit que les changements. C'est la plus rapide mais elle nécessite une table de journalisation

    CREATE MATERIALIZED VIEW LOG ON MV_Client;

    puis

    CREATE MATERIALIZED VIEW MV_Client REFRESH [ON DEMAND | ON COMMIT | START WITH ….] FAST
    AS SELECT * FROM Client;
  - COMPLETE : Oracle exécute la requête complètement :

    CREATE MATERIALIZED VIEW MV_Client REFRESH [ON DEMAND | ON COMMIT | START WITH ….] COMPLETE
    AS SELECT * FROM Client;
  - FORCE : Oracle essaie d'utiliser la méthode FAST sinon il utilise la méthode complète. C'est la méthode par défaut.