

Bases de Données

C8 - Vues - Privilèges - Optimisations

Lina Soualmia

Université de Rouen
LITIS - Équipe TIBS-CISMeF
lina.soualmia@chu-rouen.fr

4 novembre 2015


1 / 124

Vues



Vues

Une vue est

- une table virtuelle
- le résultat d'une requête que l'on nomme

Objectifs :

- Simplification des requêtes complexes
- Résolution des problèmes de confidentialité
- Garantie de l'intégrité d'une base


3 / 124

Création d'une vue

```
CREATE [OR REPLACE] [FORCE|NO FORCE]
VIEW <nom de vue> [(<listeattributs>)]
AS <clause_select>
[WITH CHECK OPTION [CONSTRAINT <nom_contrainte>]]
[WITH READ ONLY]
```

Exemple :

```
CREATE VIEW ProfMath AS
SELECT * FROM Professeurs WHERE Nom_Cours='Mathématiques';
```


4 / 124

Utilisation d'une vue

```
SELECT ... FROM <nom_de_vue> WHERE ...
```

Exemple :

```
SELECT * FROM ProfMath WHERE age>30;
```

Utilisation du nom de la vue : dans toute requête où le nom d'une table peut être spécifié (SELECT, UPDATE, DELETE, INSERT, GRANT)


5 / 124

Suppression et renommage

- Suppression :
 - ▶ DROP VIEW <nom de vue> supprime la vue mais pas les données
- Renommage :
 - ▶ RENAME <ancien nom de vue> TO <nouveau nom de vue>
- Liste des vues :
 - ▶ Consultation des tables système ALL-CATALOG, USER VIEWS et ALL VIEWS


6 / 124

Exemple de simplification de requêtes complexes

Utilisation :

```
CREATE VIEW FilmCritique (NumFilm, Titre, NomReal, NomCrit, Note)
AS SELECT F.NumFilm, Titre, I1.NomIndividu, I2.NomIndividu, Note
FROM Film F, Individu I1, Individu I2, Critique C
WHERE C.NumFilm = F.NumFilm
AND C.NumCritique = I2.NumIndividu
AND F.NumRealisateur = I1.NumIndividu;
```

```
SELECT NumFilm, Titre, NomCrit, Note
FROM FilmCritique WHERE NomReal='TOTO';
```


7 / 124

Exemple de restriction d'accès

Confidentialité :

```
CREATE VIEW PROFVIEW
AS SELECT Num_Prof, Nom_Prof, Specialite, Salaire_Actuel
FROM Professeurs;
```

Seul le créateur de Professeurs accède à Professeurs, et il ne donne l'accès aux autres, qu'à la vue.

Visibilité des informations uniquement entre 8h et 17h :

```
CREATE VIEW Professeur_Ouvrable
AS SELECT * FROM Professeurs
WHERE TO CHAR(SYSDATE, 'HH') BETWEEN '08' AND '17';
```


8 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Vérification des contraintes d'intégrité

```
CREATE VIEW CritiqueSansAvis
AS SELECT FROM Critique
WHERE Appreciation IS NULL;
```

Puis mise à jour :

```
UPDATE CritiqueSansAvis
SET Appreciation='Sans intérêt'
WHERE Note = 0;
```

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

9 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Modification d'une vue

Mise à jour ou modification (UPDATE, DELETE, INSERT) possibles si la ou les lignes concernées peuvent être retrouvées.

La vue ne doit pas contenir :

- d'opérateurs ensemblistes (UNION, EXCEPT, INTERSECT)
- d'opérateurs DISTINCT
- de fonction d'agrégation comme attribut
- de clause GROUP BY, ORDER BY, CONNECT BY
- de jointure
- de sous-requêtes
- Oracle doit être capable de déduire les modifications des tables

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

10 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Particularité d'Oracle : Condition de mise à jour

- Jointures possible si la clé est préservée dans la table résultat

```
CREATE OR REPLACE VIEW VueFilmsComplets
AS SELECT numFilm, titre,
        NumIndividu AS NumRealisateur,
        NomIndividu AS NomRealisateur,
        PrenomIndividu AS PreRealisateur
FROM Film, Individu
WHERE Realisateur = NumIndividu
```

- Mise à jour possible :

```
UPDATE VueFilmsComplets SET titre=UPPER(titre);
```

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

11 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Particularité d'Oracle : Condition de mise à jour

Mises à jour impossibles :

```
UPDATE VueFilmsComplets
SET NomRealisateur = 'LUCAS'
WHERE titre=UPPER('l''empire contre-attaque');
```

NumIndividu ne pourrait pas être clé primaire de la vue

```
INSERT INTO VueFilmsComplets (numFilm, Titre, NomRealisateur, PreReal)
VALUES (200000, UPPER ('Les Amours d''Astreeet de Celadon'), 'ROHMER', 'ERIC');
```

Les attributs n'appartiennent pas à une seule table

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

12 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Sécurité et Confidentialité d'une base

- Parmi les rôles d'un Administrateur d'une Base de Données, le rôle organisationnel concerne
 - la définition du schéma conceptuel des données
 - le partage de ces données par les utilisateurs

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

13 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Tâches de l'administrateur

- créer la BD et assurer son évolution :
 - l'administrateur a la tâche de créer la BD et ses composantes conformément à un schéma conceptuel. Il doit assurer ainsi l'évolution de cette base en modifiant, en créant ou en supprimant certaines structures.
- gérer les privilèges d'accès :
 - l'administrateur attribue et retire des privilèges d'accès aux données aux différents utilisateurs de la base de données
- assurer la sécurité et la cohérence des données :
 - l'administrateur doit mettre en place les structures et les procédures permettant de faire face à tous les incidents et de retrouver l'intégrité et la cohérence des données

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

14 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Le niveau externe : vues et droits d'accès

```
CREATE VIEW V AS (SELECT ...);
CREATE OR REPLACE VIEW V AS (requête de définition);
CREATE OR REPLACE VIEW V AS (SELECT ...) WITH GRANT OPTION;
CREATE OR REPLACE VIEW V AS (SELECT ...) WITH CHECK OPTION ;
GRANT ...;
REVOKE ...;
CREATE ROLE ...;
```

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

15 / 124

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

Le niveau externe : Vues et droits d'accès

Le niveau externe correspond à la partie visible de la base de données pour chaque utilisateur.

Dans un contexte multi-utilisateurs, un SGBD doit :

- assurer la confidentialité et l'intégrité des données. La confidentialité peut être assurée par l'utilisation des vues (views), qui agissent comme des fenêtres sur la BD;
- permettre de contrôler l'accès aux données (privilèges en lecture, modification ou suppression). Il s'agit de droits d'accès.

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Définitions

Exemples d'utilisation

Modifications

16 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Création d'une Vue mono-table :

```
CREATE OR REPLACE VIEW Profs80
AS (SELECT Numm_Prof, Nom, Spécialité, Date_Entrée
FROM Professeurs
WHERE Date_Entrée>'01-01-1980');

CREATE [OR REPLACE] VIEW Nom_Vue AS (Requête_de_définition_vue);
```

17 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Vues mono-table

```
TTITLE 'Les Professeurs'
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Professeurs;
```

Les Professeurs :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
7	Francesca		01/10/75
8	Pucette	sql	06/12/88

18 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Vues mono-table

```
TTITLE 'Les Professeurs embauchés depuis 1980'
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Prof80;
```

Les Professeurs embauchés depuis 1980 :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88

19 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion

```
INSERT INTO Profs80
VALUES (17, 'Le nouveau', 'Data WareHouse ', '25-03-2003');
```

```
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Professeurs;
```

LES PROFESSEURS :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
7	Francesca		01/10/75
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	25/03/03

20 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion

```
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Profs80;
```

Les Professeurs embauchés d

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	25/03/03

21 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion

```
UPDATE Profs80
SET Date_Entrée='01-04-2003'
WHERE Num_Prof=17;
```

LES PROFESSEURS :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03
18	Le nouveau	Data WareHouse	01/01/70

22 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion

```
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Professeurs;
```

LES PROFESSEURS :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
7	Francesca		01/10/75
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03
18	Le nouveau	Data WareHouse	01/01/70

23 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion dans la vue

```
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Profs80;
```

Les Professeurs embauchés d

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03

24 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion dans la vue

```
SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Profs80;
```

Les Professeurs embauchés d

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03

25 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour : clause WITH CHECK OPTION

La clause WITH CHECK OPTION empêche qu'un utilisateur mette à jour dans une vue des tuples non conformes à la définition de la vue.

Création avec WITH CHECK OPTION :

```
CREATE OR REPLACE VIEW Profs80
AS (SELECT Num_Prof, Nom, Spécialité, Date_Entrée
FROM Professeurs
WHERE Date_Entrée>'01-01-1980')
WITH CHECK OPTION;
```

25 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - clause WITH CHECK OPTION

```
SELECT * FROM Professeurs;
```

LES PROFESSEURS :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
7	Francesca		01/10/75
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03
18	Le nouveau	Data WareHouse	01/01/70

```
SELECT * FROM Profs80;
```

Les Professeurs embauchés depuis 1980 :

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90

26 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion dans la vue

```
INSERT INTO Profs80
VALUES (27,'Le nouveau','Data WareHouse','25-03-2003');
```

NUM_PROF	NOM	SPECIALITE	DATE_ENT
1	Bottle	poésie	01/10/70
2	Bolenov	réseau	15/11/68
3	Tonilaclasse	poo	01/10/79
4	Pastecnov	sql	01/10/75
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
7	Francesca		01/10/75
8	Pucette	sql	06/12/88
18	Le nouveau	Data WareHouse	01/01/70
17	Le nouveau	Data WareHouse	01/04/03
27	Le nouveau	Data WareHouse	25/03/03

27 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion dans la vue

NUM_PROF	NOM	SPECIALITE	DATE_ENT
5	Selector	sql	15/10/82
6	Vilplusplus	poo	25/04/90
8	Pucette	sql	06/12/88
17	Le nouveau	Data WareHouse	01/04/03
27	Le nouveau	Data WareHouse	25/03/03

L'insertion est prise en compte dans la table et dans la vue issue de cette table.

28 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Mise à jour - Insertion dans la vue

```
INSERT INTO Profs80
VALUES (28,'Le nouveau','Data WareHouse','25-04-1979');
```

```
INSERT INTO Profs80 *
ERREUR à la ligne 1 :
ORA-01402: vue WITH CHECK OPTION – violation de clause WHERE
```

NB : L'insertion n'est prise en compte ni dans la table ni dans la vue.

29 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Conditions pour les mises à jour

- La norme SQL2 a défini un ensemble de conditions que doit respecter une vue pour être modifiable
- La définition de la vue ne doit pas contenir après la clause FROM les directives JOIN, UNION, INTERSECT ou EXCEPT.
- La clause SELECT ne doit pas contenir la directive DISTINCT.
- La clause SELECT ne peut contenir que des références aux colonnes de la table source.
- Les opérateurs d'agrégats (SUM, AVG, COUNT, MIN, MAX) sont exclus.
- La clause FROM contient exactement une table ou une vue elle-même modifiable.
- La clause WHERE ne doit pas contenir de clause GROUP BY ou HAVING.

30 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Les Vues multi-tables

Création :

```
CREATE OR REPLACE VIEW Enseigner(NUMP,NOMP,SPECIALITE,NOMC,NUMC)
AS (SELECT P.Num_Prof,P.Nom,P.Spécialité,C.Nom,C.Num_Cours
FROM Professeurs P, Cours C, Charge H
WHERE C.Num_Cours=H.Num_Cours
AND H.Num_Prof=P.Num_Prof
AND C.Année=2);
```

```
SELECT * FROM Enseigner;
```

31 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

```
SELECT * FROM Enseigner;
```

NUMP	NOMP	SPECIALITE	NOMC	NUMC
1	Bottle	poésie	Sgbd	4
3	Tonilaclasse	poo	Sgbd	4
3	Tonilaclasse	poo	Analyse	5
7	Francesca		Sgbd	4
8	Pucette	sql	Sgbd	4
8	Pucette	sql	Analyse	5

```
SQL> insert into Enseigner values (37, 'Le nouveau', 'Musique', 'Danse', 6);
```

```
insert into Enseigner values (37, 'Le nouveau', 'Musique', 'Danse', 6)
*
ERREUR à la ligne 1 :
ORA-01779: Impossible modifier colonne correspondant à une table non
protégée par clé
```

32 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Vues et droits accès
Vue mono-table
Vues multi-tables

Vues multi-tables

```
INSERT INTO Enseigner
VALUES (37,'Le nouveau','Musique','Danse',6); --IMPOSSIBLE
```

Ces vues sont accessibles uniquement en lecture. Pour pallier cette insuffisance, les SGBD proposent un trigger particulier `INSTEAD OF`. Le trigger permet l'insertion, la modification et la suppression dans ce type de vues.

```
CREATE OR REPLACE TRIGGER insert_enseigner
INSTEAD OF INSERT ON Enseigner
BEGIN
    INSERT INTO Professeurs VALUES (:new.Num_Prof, :new.Nom, :new.Spécialité);
    INSERT INTO Cours VALUES (:new.Num_Cours, :new.Nom, 2);
    INSERT INTO Charge VALUES (:new.Num_Prof, :new.Num_Cours);
END;
/
```

33 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Tâches de l'administrateur

- Faut-il partager les données ?
- Faut-il attribuer/retirer des privilèges d'accès ?
- Quel type de droits d'accès à accorder aux autres utilisateurs ?

```
GRANT {privileges_systeme|rôle}
[, {privileges_systeme2|rôle2}...]
TO {utilisateur|rôle|PUBLIC} [,utilisateur2,...]
[WITH GRANT OPTION];
```

```
REVOKE {privileges_systeme|rôle}
[, {privileges_systeme2|rôle2}...]
FROM {utilisateur|rôle|PUBLIC} [,utilisateur2,...];
```

35 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Droits d'accès

- Attribuer des privilèges :

```
GRANT privilège|ALL TO utilisateur|PUBLIC[WITH GRANT OPTION];
```

- Retirer des privilèges :

```
REVOKE privilège|ALL FROM utilisateur|PUBLIC;
```

37 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Rôles

- Un rôle est un groupement de privilèges.
- Un utilisateur est un nom de compte Oracle.
- `PUBLIC` permet de transmettre les droits à tous les utilisateurs associés à l'espace de stockage (tablespace) concerné.
- `WITH GRANT OPTION` permet de ré-attribuer au(x) bénéficiaires(s) les privilèges définis dans `GRANT`.

39 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Tâches de l'administrateur

- Création des utilisateurs :

```
CREATE USER NomUser IDENTIFIED BY MotPasseUser
DEFAULT TABLESPACE NomBD
TEMPORARY TABLESPACE tmp NomBD ;
```

- Créer un rôle et un profil :

```
GRANT CONNECT TO NomUser;
GRANT RESOURCE TO NomUser;
GRANT SEL_CATALOG_ROLE TO NomUser;
```

- Les droits connect et resource permettent de faire de nombreux traitements sur Oracle (créer des tables, des index, des vues, des triggers, du pl/sql etc...,)
- le droit `sel_catalog_role` est très important pour pouvoir consulter le dictionnaire de données d'Oracle et faire le suivi des optimisations.

34 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Droits d'accès

Il existe deux familles de permissions :

- les permissions système : pour Oracle, plusieurs dizaines de privilèges sont disponibles
 - Par exemple, la création d'utilisateurs (`CREATE USER`),
 - la création et la suppression de tables (`CREATE TABLE` / `DROP TABLE`),
 - la création d'espace disque (`CREATE TABLESPACE`),
 - la sauvegarde de table (`BACKUP ...`),
 - la création de rôles (`CREATE ROLE`)...
- les permissions objet qui permettent de donner des droits sur les données elles mêmes.

36 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Droits d'accès

- Attribuer des privilèges système sous Oracle :

```
GRANT {privileges_systeme|rôle}
[, {privileges_systeme2|rôle2}...]
TO {utilisateur|rôle|PUBLIC} [,utilisateur2,...]
[WITH GRANT OPTION];
```

- Les principales options possibles des privilèges système d'Oracle sont :
 - grant connect
 - grant create trigger
 - grant create procedure
 - grant create session
 - grant create view
 - grant unlimited tablespace

38 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Rôles

- Attribuer des privilèges objet sous Oracle : (syntaxe simplifiée)

```
GRANT {privilège1 [(colonne11,[colonne12,...])]|rôle}
[, {privilège2[(colonne21,...)]|rôle2}...]
ON SchemaDeLobjet
TO {utilisateur|rôle|PUBLIC} [,utilisateur2...]
[WITH GRANT OPTION];
```

- Les principales options possibles des privilèges objet d'Oracle sont :

▸ ALTER, SELECT, DELETE, UPDATE, RENAME, LOCK, ...

40 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Rôles

- Retirer des privilèges objet sous Oracle : (syntaxe simplifiée)

```
REVOKE {privilège1 [(colonne1, [colonne2, ...]) | rôle}  
[, {privilège2 [(colonne21...)] | rôle2} ...]  
ON SchemaDeLobjet  
FROM {utilisateur | rôle | PUBLIC} [, utilisateur2 ...]  
[CASCADE CONSTRAINTS];
```

41 / 124

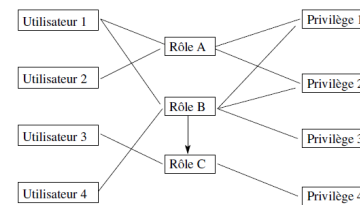
Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Les rôles sous Oracle

Le principe d'un rôle Oracle (ROLE) est de créer un groupement nommé de privilèges. Celui-ci peut être accordé à d'autres utilisateurs.



La commande SQL qui permet de créer un rôle est :

```
CREATE ROLE LeNomDuRôle;  
GRANT LePrivilege ON Lobjet TO LeRôle;
```

42 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Les profils sous Oracle

- Un profil est un ensemble de ressources limitées, à attribuer aux utilisateurs, des caractéristiques physiques telles que :
 - Un pourcentage d'utilisation dans un tablespace ;
 - Un nombre de connexions ;
 - Une priorité d'exécution des requêtes etc.

43 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Les profils sous Oracle

Exemple : Créer un profil et l'affecter à un utilisateur :

```
CREATE PROFILE profexpl  
LIMIT SESSION_PER_USER 1  
IDLE_TIME 120  
CONNECT;
```

Affecter le profil à l'utilisateur toto

```
ALTER USER toto PROFILE profexpl;
```

44 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Rôles et utilisateurs

Les principaux rôles, prédéfinis par Oracle et attribués aux utilisateurs SYSTEM et SYS, sont :

Rôle	Privilège système
CONNECT	CREATE SESSION, CREATE TABLE, ...
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, ...
DBA	Tous les privilèges

45 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Sécurité des données

- Gestion des privilèges d'accès
- Gestion des transactions
- Résolution des conflits d'accès

46 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Sécurité des données - Gestion des privilèges d'accès

Privilèges sur les tables, les vues et les index

- Accorder des privilèges :

```
GRANT privilege(s)  
ON Nom_de_Table [(Nom_de_Colonne,...)] ON Nom_de_Vue  
TO utilisateur(s) | TO PUBLIC  
[WITH GRANT OPTION];
```

- Retirer des privilèges :

```
REVOKE privilege(s)  
ON Nom_de_Table [(Nom_de_Colonne,...)] ON Nom_de_Vue  
FROM utilisateur(s) | FROM PUBLIC;
```

- Les privilèges sont :

- ALL, SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER REFERENCE, CREATE

47 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

L'instruction GRANT

- Celui qui crée une table, une vue ou un index possède tous les privilèges sur cet objet
- L'instruction GRANT lui permet d'autoriser d'autres personnes à l'utiliser.

48 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
OptimisationDroits d'accès
Rôles
Sécurité

Exemple 1 : Permettre à l'utilisateur 'Etudiant1' de lire et mettre à jour la table Eleves

```
GRANT SELECT, UPDATE ON Eleves TO Etudiant1;
```

L'utilisateur CMOI a créé la table Eleves. La lecture de celle-ci se fait par :

```
SELECT * FROM Eleves;
```

L'utilisateur Etudiant1 doit la lire avec :

```
SELECT * FROM CMOI.Eleves;
```



49 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

L'instruction GRANT

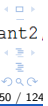
Exemple 2 : Permettre aux utilisateurs 'Etudiant1' et 'Etudiant2' de lire et mettre à jour la table Eleves.

```
GRANT SELECT, UPDATE ON Eleves TO Etudiant1, Etudiant2;
```

UPDATE est généralement la seule commande (parfois aussi SELECT), pour laquelle on peut spécifier une liste des attributs, et donc empêcher toute modification sur les autres.

Exemple 3 : Permettre aux utilisateurs 'Etudiant1' et 'Etudiant2' de lire et mettre à jour l'attribut poids de la table Eleves.

```
GRANT SELECT, UPDATE ON Eleves(poids) TO Etudiant1, Etudiant2;
```



50 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

L'instruction GRANT

Exemple 4 : Permettre à tous les utilisateurs de lire la table Activites.

```
GRANT SELECT ON Activites TO PUBLIC;
```

Exemple 5 : Permettre à un utilisateur privilégié SuperUser de tout faire sur la table Resultats

```
GRANT ALL ON Resultats TO SuperUser;
```



51 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

L'instruction GRANT

Exemple 6 : Permettre à un utilisateur privilégié SuperUser de créer des index sur la table Professeurs

```
GRANT INDEX ON Professeurs TO SuperUser;
```

Exemple 7 : Permettre à un utilisateur privilégié SuperUser de modifier la structure de la table Professeurs

```
GRANT ALTER ON Professeurs TO SuperUser;
```

Remarque : Les 2 derniers privilèges (INDEX et ALTER) ne sont pas autorisés pour une vue.



52 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Option WITH GRANT OPTION

Jusqu'à présent, ceux qui recevaient un privilège ne pouvaient le céder à d'autres. Cette faculté existe avec l'option WITH GRANT OPTION.

Exemple 8 : Donner à l'utilisateur 'Etudiant1' le droit de lire et mettre à jour la table Eleves, en lui permettant de redistribuer ce privilège à d'autres.

```
GRANT SELECT,UPDATE ON Eleves TO Etudiant1 WITH GRANT OPTION;
```

Un utilisateur ne peut distribuer de privilèges que sur les objets qu'il détient ou sur les objets pour lesquels il a reçu des privilèges avec l'option WITH GRANT OPTION



53 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Option WITH GRANT OPTION

```
GRANT SELECT ON T_CHAMBRE TO Dubois;
```

Autorise Dubois à lancer des ordres SQL SELECT sur la table T_CHAMBRE. (Notez l'absence du mot TABLE).

```
GRANT SELECT ON TABLE T_CMAMBRE TO Dufour WITH GRANT OPTION;
```

Autorise Dufour à lancer des ordres SQL SELECT sur la table T_CHAMBRE mais aussi à transmettre à tout autre utilisateur les droits qu'il a acquis dans cet ordre



54 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Option WITH GRANT OPTION

```
GRANT SELECT, UPDATE ON TABLE T_CHAMBRE TO PUBLIC;
```

Autorise tous les utilisateurs présents et à venir à lancer des ordres SQL SELECT et UPDATE sur la table T_CHAMBRE



55 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Option WITH GRANT OPTION

Durand lance l'ordre suivant :

```
GRANT ALL PRIVILEGES ON TABLE T_CHAMBRE TO Dubois;
```

Ce qui autorise Dubois à lancer sur la table T_CHAMBRE, les mêmes ordres SQL, que ceux autorisés à Durand (SELECT, INSERT, DELETE).

On parle alors d'héritage de droits c'est à dire que l'utilisateur doté de ces droits peut à nouveau les céder à un ou plusieurs autres utilisateurs.



56 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Droits d'accès Rôles Sécurité

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

L'utilisateur peut toujours reprendre un privilège qu'il a donné à un user.

Exemple 9 : Reprendre le privilège INDEX accordé à l'utilisateur SuperUser sur la table Professeurs

```
REVOKE INDEX ON Professeurs FROM SuperUser;
```

Exemple 10 : Reprendre le privilège UPDATE accordé à l'utilisateur Etudiant1 sur la table Professeurs

```
REVOKE UPDATE ON Professeurs FROM Etudiant1;
```

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

57 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

Exemple 11 : Donner tous les privilèges, sauf ALTER, sur la table Eleves, à tous les utilisateurs

```
GRANT ALL ON Eleves TO PUBLIC;
```

```
REVOKE ALTER ON Eleves FROM PUBLIC;
```

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

58 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

Exemple 12 : Ne permettre l'accès à la table Resultats qu'à SuperUser

```
REVOKE ALL ON Resultats FROM PUBLIC;
```

```
GRANT ALL ON Resultats TO SuperUser;
```

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

59 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

SQL respecte les règles suivantes :

- Si un utilisateur reçoit de plus d'une personne le privilège d'accéder à une table :
 - il pourra continuer à y accéder jusqu'à ce que son privilège lui soit retiré par tous ceux qui le lui ont donné (les privilèges identiques sont donc cumulés).
- Enlever un privilège à un utilisateur revient à l'enlever à toutes les personnes qui l'ont reçu de ce dernier.

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

60 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

```
REVOKE SELECT ON T_CHAMBRE FROM Dubois;
```

Supprime le privilège de sélection de la table T_CHAMBRE attribué à Dubois

```
REVOKE INSERT, DELETE ON TABLE T_CHAMBRE FROM Duval,Dubois;
```

Supprime les privilèges d'insertion et de suppression de la table T_CHAMBRE attribué à Duval et Dubois précédemment, mais pas celui de mise à jour (UPDATE)

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

61 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

L'instruction REVOKE

```
REVOKE GRANT OPTION FOR SELECT ON TABLE T_CHAMBRE FROM Dufour;
```

Supprime la possibilité pour Dufour de transmettre le privilège de sélection sur la table T_CHAMBRE

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

62 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Privilèges et vues

En combinant la notion de vue et une gestion adéquate des privilèges, **on atteint une sécurité presque totale des données.**

Exemple 13 : Permettre à une catégorie d'utilisateurs (ou à tous les utilisateurs) d'accéder en lecture à l'ensemble de la table Profs, sauf aux attributs Date_Entree, Der_Promo, Salaire_Base, Salaire_Actuel

```
CREATE OR REPLACE VIEW Profs(NomeroPrf, NomPrf, SpecialitePrf) AS (SELECT Num_Prof, Nom, Spécialité FROM Professeurs);
```

```
GRANT SELECT ON Profs TO PUBLIC;
```

On prendra soin de ne pas donner aux utilisateurs le privilège d'accéder directement à la table Profs. Par contre, le créateur d'une vue doit toujours disposer du privilège SELECT sur toutes les tables qui entrent dans la définition de la vue.

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

63 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Privilèges et vues

Exemple 14 : Le contrôle des informations critiques peut aussi se faire par l'intégration du nom de l'utilisateur créateur de l'objet dans la condition de sélection.

```
CREATE OR REPLACE VIEW MesCours AS (SELECT Nom, Année FROM Cours WHERE Num_Cours IN (SELECT Num_Prof FROM Charge WHERE Num_Prof IN (SELECT Num_Prof FROM Professeurs WHERE Nom = Utilisateur_lambda))));
```

```
GRANT ALL ON MesCours TO PUBLIC;
```

Le contenu de la vue MesCours est l'ensemble des cours de l'utilisateur.

Lina SoualmiaBases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

64 / 124

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Droits d'accès
Rôles
Sécurité

Optimisations

- Optimisations logiques
- Optimisations physiques



66 / 124

Optimisation d'une Application ?

- Quand décide-t-on d'optimiser ?
- Optimiser quoi ?
- Qui participe, et Comment ?
 - Type de l'application
 - Intérêt de l'application pour l'entreprise



67 / 124

Quand ?

- Application Batch
 - L'application n'est pas disponible à partir d'une certaine heure
- Application transactionnelle
 - L'utilisateur constate que le temps de réponse est inacceptable (Agence de voyage, application interne, etc.)



68 / 124

Qui participe et pourquoi ?

- Adminstrateurs de DB, Développeurs, Ingénieurs Système (IS), Chef de projet
 - Définition de la puissance de la machine (Nombre de processeurs, Capacité mémoire, Espace disque nécessaire en fonction du type de l'application)
 - Type du SGBD (Oracle, DB2, etc.)
 - Type de l'application
 - Langage de programmation utilisé (Java, PL/SQL, etc.)



69 / 124

Rôle d'un administrateur de BD (DBA)

- Rôle organisationnel :
 - Définition du schéma conceptuel des données
 - Partage de ces données par les utilisateurs
- Rôle technique
 - Mise en oeuvre du schéma
 - Partage à l'aide des capacités techniques du SGBD



70 / 124

Rôle technique du DBA

- Installation du SGBD et des outils associés
- Création de la BD et de ses composants conformément à un schéma conceptuel
- Surveillance de son évolution en modifiant, en créant ou en supprimant certaines structures
- Gestion des privilèges d'accès
- Attribution et retrait de privilèges d'accès aux données aux différents utilisateurs de la base de données



71 / 124

Rôle technique du DBA

Amélioration les performances

- Choix de l'implantation optimale des données de façon à obtenir les meilleures performances
- Identification et prise en compte des utilisations qui seront faites des données

Surveillance de la sécurité et la cohérence des données

- Mise en place de structures et de procédures permettant
 - de faire face à tous les incidents
 - de retrouver l'intégrité et la cohérence des données



72 / 124

Rôle technique DBA

- Échange les données entre la BD et le monde extérieur :
 - Surveillance de l'intégration des données en provenance d'autres applications ou BD
 - Migration des données de la base vers d'autres applications ou B
- Outils à utiliser : SQL DBA, SQL Loader, SQL Plus etc...



73 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Sur quoi doit-on focaliser les efforts d'optimisation ?

Partie 1 :

- Conception des systèmes d'informations et optimisation des applications
 - Lors de la conception des systèmes d'informations (si pas trop tard)
 - Optimisation des applications

Partie 2 :

- Présentation des outils pour assurer le suivi de la base et garantir sa performance
- Optimisation de la mémoire
 - Optimisation des entrées/sorties disque
 - Identifier les contentions dans la base de données



75 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation des applications

- L'expérience montre que 80% des problèmes de performances des applications, sont résolus par une optimisation des requêtes et procédures SQL
- Ordonnancer les Batches et éviter leur exécution pendant des heures où l'utilisation des machines est intense
- Dispatcher les Batches les plus consommateurs en puissance machine à des heures différentes



77 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Outils pour assurer le suivi de la base et garantir sa performance

Identifier les contentions dans la base de données :

- Etudier les locks et les wait events au niveau de la base de données,
- et les éliminer si possible



79 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Qu'est-ce qui reste à faire ?

- Les besoins en puissance machine définis
- Les acteurs administrateurs BD, (IS), développeurs, etc. au courant du projet (impliqués, responsable de la réussite du projet,...)
- Optimisation de la base et de l'application ?



74 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Partie 1 : Conception des systèmes d'information et optimisation des applications

Lors de la conception des systèmes d'information :

- Système non performant :
 - résultat d'une mauvaise définition du modèle conceptuel
 - Modèle conceptuel : au moins sous la 3ème forme Normale, sauf dans quelques cas (choix volontaire) où la dé-normalisation apporte une certaine performance au système d'information (DataWarehouse)

Lors de la conception : tenir compte de l'accès aux données

- Analyse de la répartition des données :
 - réplication de données (sur une ou plusieurs bases, etc.),
 - agrégation des tables, pour les systèmes décisionnels etc.



76 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Partie 2 : Outils pour assurer le suivi de la base et garantir sa performance

Optimisation de la mémoire :

- déterminer la bonne taille des buffers de la base (shared_pool, buffer cache, log buffer, etc)

Optimisation des entrées/sorties disque :

- bien dimensionner les fichiers de la base de données
- et placer dans des disques prévus pour le type d'application, pour assurer un temps de réponse acceptable des requêtes adressées à la base.



78 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation des applications

- L'expérience montre que l'optimisation des requêtes SQL résout la majorité des problèmes de performances des applications

Objectifs de l'optimisation

- Clarifier et situer les différents paramètres internes des SGBD (relationnel ; objet-relationnel) afin d'améliorer leurs performances.
- Réécrire les codes SQL ; PL/SQL etc..
- Restructurer les données, indexer les données, créer des vues matérialisées
- Partager les données, dupliquer les données sur plusieurs disques, créer des partitions sur les données



80 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Objectifs de l'optimisation

Intervenants :

- Administrateurs de Bases de Données
- Programmeurs d'Applications sur SGBD

Objectifs :

- Optimiser un système existant et connaître les impacts de certains paramètres en fonction du type d'application sur le système
- Choisir un SGBD en ayant comme contrainte les critères de performance



81 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Augmentation des performances

Pour augmenter les performances trois grandes solutions apparaissent :

- 1 Première solution (d'ordre logique) : optimiser les schémas conceptuel et logique pour qu'ils collent aux applications
- 2 Deuxième solution (d'ordre physique) : optimiser les paramètres internes du SGBD
- 3 Autre solution (de type matérielle) : augmenter la puissance machine ou utiliser des ordinateurs spéciaux dédiés à la gestion des données



82 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation logique de la Base de Données

- Schéma Conceptuel de Données (SCD) :
 - ▶ obtenu à la phase d'analyse
 - ▶ un ensemble d'Entités et d'Associations ou un ensemble de Classes selon le formalisme utilisé
- En E/A, transformation en schéma relationnel (Schéma Logique de Données SLD), qui permet
 - ▶ Implantation du SCD dans une BD relationnelle
 - ▶ Exploitation par le SGBD et les modules de programmation



83 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation logique de la Base de Données

Optimisation du Schéma Relationnel :

- Normalisation : processus permettant de s'assurer de la Bonne conception du SLD
- non-redondance de ses données
- Cadre formel pour effectuer la normalisation :
 - ▶ les dépendances fonctionnelles
 - ▶ les formes normales (1ère FN, 2ème FN, 3ème FN, FNBC, 4ème FN, ...)



84 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Normalisation

Avant normalisation :

- Relation1(Attribut1, Attribut2, ...)

Décomposition de la relation Relation1 en deux ou plusieurs relations qui contiennent moins d'anomalies de mises à jour (peu ou pas du tout) : on parle de

- Décomposition sans perte d'information
- Décomposition sans perte de dépendance fonctionnelle (DF)

Après normalisation :

- Relation11(Attribut11, Attribut12, ...)
- Relation12(Attribut21, Attribut22, ...)



85 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

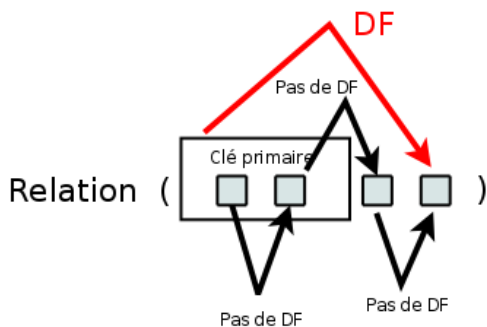
Normalisation

- Les relations obtenues doivent être en 3ème Forme Normale, au moins : bon rapport redondance/espace occupé
- Clé Primaire
- Pas de DF entre les attributs de la clé primaire;
- Pas de DF entre un sous-ensemble d'attributs de la clé primaire et les attributs qui n'appartiennent pas à la clé primaire
- Pas de DF entre les attributs qui n'appartiennent pas à la clé primaire



86 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL



87 / 124

Lina Soualmia	Bases de Données
---------------	------------------

Exemple

- Soit la relation :
 - ▶ ADRESSES (Ville, Numero, Rue, CodePostal)
- Décomposition de la relation ADRESSES en deux relations CPR et CPV :
 - ▶ CPR (CodePostal, Rue)
 - ▶ CPV (CodePostal, Ville)



88 / 124

Lina Soualmia	Bases de Données
---------------	------------------

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

ADRESSES

VILLE

Paris

6

Rue de la Rosière

75015

Paris

16

Rue de la Rosière

75015

Paris

51

Rue des Entrepreneurs

75015

Paris

55

Rue des Entrepreneurs

75015

Paris

8

Avenue des Champs Elysées

75008

Epina sur Seine

23

Boulevard Foch

93800

CPR

CODE POSTAL

75015

RUE

Rue de la Rosière

75015

75015

Rue des Entrepreneurs

75008

Avenue des Champs Elysées

93800

Boulevard Foch

CPV

CODE POSTAL

75008

75015

93800

VILLE

Paris

Paris

Epina sur Seine

89 / 124

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Un client a une adresse :

- SLD version 1
 - CLIENT (CodeCli, NomCli, PrenomCli, Numero, Rue, CodePostal, Ville)
- SLD version 2
 - CLIENT (CodeCli, NomCli, PrenomCli, Numero, AdrCli*)
 - ADRCPR (AdrCli, Rue, CodePostal*)
 - ADRCPV (CodePostal, Ville)

```
SELECT * FROM Client;  
  
SELECT * FROM CLIENT, ADRCPR, ADRCPV  
WHERE CLIENT.AdrCli=ADRCPR.AdrCli  
AND ADRCPR.CodePostal=ADRCPV.CodePostal;
```



Bilan

Un schéma relationnel normalisé

- contiendra donc un nombre plus important de relations (de tables)
- pourra pénaliser certaines requêtes, qui seront obligées de procéder à des jointures plus nombreuses sur la base

Or

- La jointure est une opération très coûteuse
- Les opérations binaires (qui portent sur deux tables) peuvent être coûteuses



Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Redondance calculée

- Technique d'optimisation des requêtes d'interrogation
- Introduction volontaire de redondances dans le schéma logique relationnel
- Cette redondance est dite calculée car elle tient compte
 - des besoins des modules de traitement des données
 - de leurs exigences en terme de temps de réponse



Redondance calculée

- Deux méthodes de réalisation de la redondance calculée :
 - le stockage des données déductibles
 - la dé-normalisation



Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Stockage des données déductibles

Données déductibles

- les résultats des requêtes les plus fréquentes
- des statistiques historiques
- des données issues de calculs complexes



Redondance calculée : stockage des données déductibles

- Dans les trois cas, stockage des données, physiquement dans la base (sous la forme de Tables/Vues ou de Colonnes)
- Avantage du stockage physique : éviter leur re-génération en cas de besoin
- Inconvénients de la redondance :
 - Place occupée par les données redondantes
 - Nécessité de traitements supplémentaires pour les opérations de mises à jour
 - Risque d'incohérence



Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues

Sécurité et confidentialité

Gestion des utilisateurs

Optimisation

Optimisation logique

Redondance calculée

Redondance calculée

Optimisation de ordres SQL

Exemple de requête fréquente

- Base de données SportAct : Adhérents à des centres sportifs
 - CENTRE(NumC, NomC, VilleC, Coutinsc)
 - ESTMEMBRE(NumA, NumC*, Dateinsc)
- Requête fréquente : nombre d'inscrits dans un centre c ?

```
SELECT C.NumC, C.NomC, COUNT(E.NumA)  
FROM CENTRE C, ESTMEMBRE E  
WHERE C.NumC = E.NumC  
AND C.NumC = c  
GROUP BY C.NumC;
```



Exemple de requête fréquente

- Création d'une nouvelle colonne NbrInscr dans la table CENTRE :
 - CENTRE(NumC,NomC,VilC,Coutinsc,NbrInscr)
- Nouvel ordre SQL :

```
SELECT NumC, NomC, NbrInscr  
FROM CENTRE WHERE NumC = c;
```

- Suppression de 2 opérations très coûteuses :
 - Une jointure en moins
 - Un groupement en moins



97 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple de requête fréquente

Impact sur le reste de la base de données :

- Prise en compte de cette nouveauté par les opérations de mises à jour de la table ESTMEMBRE
- Modification de la colonne NbrInscr à chaque insertion ou suppression d'inscription
- La cohérence est affectée si on ne le fait pas



98 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple de requête fréquente

Chaque opération INSERT DELETE implique une opération UPDATE

- INSERT INTO ESTMEMBRE VALUES ... doit être accompagnée par :
 - UPDATE CENTRE SET NbrInscr = NbrInscr + 1 ;
- DELETE FROM ESTMEMBRE WHERE ... doit être accompagnée par
 - UPDATE CENTRE SET NbrInscr = NbrInscr - 1 ;



99 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple de Statistiques : Historique

- Requête : Nombre d'inscrits par centre et par mois ?
- Objectif : éviter le re-calcul à chaque fois du nombre d'inscrits par centre et par mois
- Solution : création d'une nouvelle table/vue :
 - StatMensInsc (NumC, Mois,NbrInscr)
- Nécessité d'un nouveau traitement mensuel pour l'actualiser avec les statistiques du mois écoulé



100 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Résultat de calcul complexe

Soit la base de données :

- CLIENT(CodeCli,NomCli,...)
- ARTICLE(CodeArt,LibArt,...,PrixArt)
- COMMANDE(NumCom,CodeCli*,DateCom)
- DETAILCOM(NumCom*,CodeArt*,QteComDee)
- Requête : Montant de la commande ?



101 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Résultat de calcul complexe

Obtention du montant de la commande :

- Gain d'opération avec la création de la colonne Montant dans la table COMMANDE

```
SELECT C.NumCom, SUM(D.QteComDée*A.PrixArt) AS Montant  
FROM COMMANDE C, ARTICLE A, DETAILCOM D  
WHERE C.NumCom=D.NumCom AND D.CodeArt=A.CodeArt  
GROUP BY C.NumCom ;
```

- COMMANDE(NumCom,CodeCli,DateCom,Montant)

```
SELECT NumCom, Montant FROM COMMANDE ;
```



102 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Redondance calculée : dé-normalisation

- Autre méthode d'optimisation des interrogations
- Transformer, si besoin est, une table de la 3ème FN en 2ème FN ou en la 1ère FN
- Objectif : éviter des jointures successives pouvant être coûteuses en performances



103 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple : Redondance calculée : dé-normalisation

- Soit la base
 - ARTICLE(CodeArt,LibArt,...,PrixArt)
 - COMMANDE(NumCom,CodeCli*,DateCom)
 - DETAILCOM(NumCom,CodeArt*,QteComDee,PrixArt)
- DETAILCOM passe de la 3ème FN à la 1ère FN
- Requête : montant de la commande ?

```
SELECT NumCom, SUM(QteComDee*PrixArt) AS Montant  
FROM DETAILCOM GROUP BY NumCom;
```



104 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Redondance calculée : dé-normalisation

- Inconvénients de la dé-normalisation : identiques à ceux du stockage des données déductibles
 - Place supplémentaire
 - Pénalisation des traitements d'insertion et de suppression
 - Risque d'incohérence
- Nécessite la création de déclencheurs (triggers)



105 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation des ordres SQL

- Traduction d'une requête exprimée en langage naturel en un ensemble d'ordres SQL
- Plusieurs agencements sont souvent possibles : au niveau de l'ordre dans lequel les conditions sont exprimées dans la clause WHERE :
 - WHERE condition1 AND condition2
 - WHERE condition2 AND condition1



106 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation des ordres SQL

- Traduction d'une requête exprimée en langage naturel en un ensemble d'ordres SQL
- Plusieurs agencements sont souvent possibles :
 - au niveau des opérations algébriques : Possibilité d'utiliser plusieurs méthodes différentes pour exprimer la même chose
 - La jointure : possibilité d'exprimer l'opération de plusieurs façons différentes
 - $R = \text{Jointure}(R1, R2, \{R1.A=R2.A\}) = \text{Jointure}(R2, R1, \{R2.A=R1.A\})$
 - SELECT * FROM R1,R2 WHERE R1.A=R2.A;
 - SELECT * FROM R2,R1 WHERE R2.A=R1.A;



107 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

- Méthode prédicative :

```
SELECT * FROM R1,R2 WHERE R1.A=R2.A;
```

- Méthodes ensemblistes :

```
SELECT * FROM R1 WHERE A IN (SELECT A FROM R2);
```

```
SELECT * FROM R1 WHERE A =ANY (SELECT A FROM R2 );
```

```
SELECT * FROM R1 WHERE EXISTS (SELECT * FROM R2 WHERE R1.A=R2.A);
```

```
SELECT * FROM R1 WHERE 0 < (SELECT COUNT(*) FROM R2 WHERE R1.A=R2.A)
```



108 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple de durée d'exécution - SQLPlus

- SET TIMING ON
 - Exécution de la requête; SELECT ... FROM ... : Oracle donne le temps écoulé
- Méthode prédicative :

```
select * from all_tables t1, all_constraints t2
      where t1.table_name = t2.table_name;
```

```
temps CPU: 1 091 480 unités      451 100 unités
```

```
select * from all_constraints t1, all_tables t2
      where t1.table_name = t2.table_name;
```

```
temps CPU: 976 790 unités      690 690 unités
```



109 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Exemple de durée d'exécution - SQL PLus

- Méthodes ensemblistes :

```
select * from all_constraints where table_name
      in (select table_name from all_tables);
```

```
temps CPU: 138 350 unités      139 230 unités
```

```
select * from all_constraints where table_name
      = any (select table_name from all_tables);
```

```
temps CPU: 139 630 unités      139 890 unités
```

```
select * from all_constraints t1 where exists
      (select * from all_tables t2 where
       t1.table_name = t2.table_name);
```

```
temps CPU: 177 580 unités      181 360 unités
```



110 / 124

Lina Soualmia	Bases de Données
Vues Sécurité et confidentialité Gestion des utilisateurs Optimisation	Optimisation logique Redondance calculée Redondance calculée Optimisation de ordres SQL

Optimisation des ordres SQL

- Traduction d'une requête exprimée en langage naturel en un ensemble d'ordres SQL :
- Plusieurs agencements sont souvent possibles
- Possibilité d'utiliser plusieurs méthodes différentes pour exprimer la même chose
- La différence :
 - R = Difference (R1, R2)
 - SELECT * FROM R1 MINUS SELECT * FROM R2 ;



111 / 124

Lina Soualmia	Bases de Données
---------------	------------------

Expression de la différence

```
SELECT A FROM R1 MINUS SELECT A FROM R2 ;
```

```
SELECT A FROM R1 WHERE A NOT IN (SELECT A FROM R2) ;
```

```
SELECT A FROM R1 WHERE NOT EXISTS (SELECT A FROM R2 WHERE R1.A=R2.A)
```

```
SELECT A FROM R1 WHERE 0 = (SELECT COUNT(*) FROM R2 WHERE R1.A=R2.A)
```



112 / 124

Lina Soualmia	Bases de Données
---------------	------------------

Utilisation des vues

- Mauvais code :

```
SELECT e.*
FROM EMP e
WHERE e.salary > (SELECT AVG (salary)
                  FROM EMP i
                  WHERE i.depid = e.depid);
```

- Bon code

```
SELECT e.*
FROM EMP e, (SELECT i.depid DEP, avg (i.salary ) SAL
              FROM EMP i GROUP BY i.depid) EMP_VUE
WHERE e.depid= EMP_VUE.depid AND e.salary > EMP_VUE.SAL;
```

113 / 124

Vues matérialisées

Présentation des vues matérialisées :

- Création d'une vue physique d'une table
- Duplication des données
- Définition de vues matérialisées à partir de tables, de vues, et de vues matérialisées
- Possibilité de décalage entre la table maître et la vue matérialisée : gestion de la fraîcheur des données de la vue matérialisée (refresh)

114 / 124

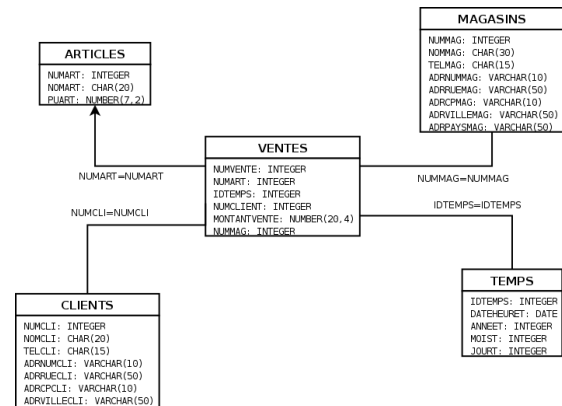
Vues matérialisées

Utilisation des vues matérialisées :

- Optimisation/amélioration des performances
- Ré-écriture des requêtes portant sur des tables
 - SELECT particulièrement complexe ou lourd
 - réplication de table

115 / 124

Exemple :



116 / 124

Exemple d'utilisation de vues matérialisées

- Soit la requête Q : Total des ventes des magasins en France ou en Belgique -pour la période de 2001 à 2003- par ville et année

```
SELECT ADRVILLEMAG, ANNEET, SUM(MONTANTVENTE) AS Montant
FROM VENTES, MAGASINS, TEMPS
WHERE VENTES.NUMMAG=MAGASINS.NUMMAG
AND VENTES.IDTEMPS=TEMPS.IDTEMPS
AND (UPPER(MAGASINS.ADRPAYSMAG)='FRANCE' OR UPPER (MAGASINS.ADRPAYSMAG)='BELGIQUE'
AND TEMPS.ANNEET BETWEEN 2001 AND 2003
GROUP BY ADRVILLEMAG, ANNEET ;
```

117 / 124

Exemple d'utilisation de vues matérialisées

- V1 : Total des ventes des magasins à partir de 2002, par ville et année

```
CREATE MATERIALIZED VIEW v1 (Ville, Annee, Montant1)
AS (SELECT ADRVILLEMAG, ANNEET, SUM(MONTANTVENTE)
FROM VENTES, MAGASINS, TEMPS
WHERE VENTES.NUMMAG=MAGASINS.NUMMAG
AND VENTES.IDTEMPS=TEMPS.IDTEMPS
AND TEMPS.ANNEET >= 2002
GROUP BY ADRVILLEMAG, ANNEET);
```

118 / 124

- V2 : Total des ventes des magasins en France par ville, année et mois

```
CREATE MATERIALIZED VIEW v2(Ville, Annee, Mois, Montant2)
AS (SELECT ADRVILLEMAG, ANNEET, MOIST, SUM(MONTANTVENTE)
FROM VENTES, MAGASINS, TEMPS
WHERE VENTES.NUMMAG=MAGASINS.NUMMAG
AND VENTES.IDTEMPS=TEMPS.IDTEMPS
AND UPPER (MAGASINS.ADRPAYSMAG)= 'FRANCE '
GROUP BY ADRVILLEMAG, ANNEET, MOIST);
```

119 / 124

- V3 : Total des ventes des magasins en Belgique, avant 2001, par ville, année et mois

```
CREATE MATERIALIZED VIEW v3(Ville, Annee, Mois, Montant3)
AS (SELECT ADRVILLEMAG, ANNEET, MOIST, SUM(MONTANTVENTE)
FROM VENTES, MAGASINS, TEMPS
WHERE VENTES.NUMMAG=MAGASINS.NUMMAG
AND VENTES.IDTEMPS=TEMPS.IDTEMPS
AND UPPER (MAGASINS.ADRPAYSMAG)= 'BELGIQUE '
and TEMPS.ANNEET <= 2001
GROUP BY ADRVILLEMAG, ANNEET, MOIST);
```

120 / 124

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

- Soit la requête Q' qui utilise des vues matérialisées :

```
SELECT ADRVILLEMAG, annee , Montant1
FROM v1 ,
      (SELECT DISTINCT ADRVILLEMAG
FROM MAGASINS
WHERE UPPER (MAGASINS.ADRPAYSMAG)= 'FRANCE '
OR UPPPER (MAGASINS.ADRPAYSMAG)= 'BELGIQUE ' ) s1
WHERE v1.ville= s1.ADRVILLEMAG AND annee <= 2003
UNION
SELECT ville, annee, SUM(Montant2)
FROM v2 WHERE annee=2001
GROUP BY ville, annee
UNION
SELECT s2.ADRVILLEMAG, Annee, SUM(Montant3)
FROM v3 ,
      (SELECT DISTINCT ADRVILLEMAG FROM MAGASINS) s2
WHERE v3.ville = s2.ADRVILLEMAG
AND annee = 2001
GROUP BY s2.ADRVILLEMAG, annee;
```

121 / 124

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Utilisation de vues matérialisées

Avantages :

- Simple réécriture de requêtes et exécution de requêtes imbriquées
- Lorsqu'il n'est pas nécessaire de disposer des données en temps réel, économie d'interrogation
 - Exemple : analyse de chiffres économiques de la veille
 - données figées
 - intérêt à stocker des résultats (qui ne changeront pas)
 - Utilisation dans les datawarehouse : fouille de données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Lina Soualmia

Bases de Données

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Vues
Sécurité et confidentialité
Gestion des utilisateurs
Optimisation

Optimisation logique
Redondance calculée
Redondance calculée
Optimisation de ordres SQL

Utilisation de vues matérialisées

Inconvénients :

- Résultat de la requête stocké et valable à un instant t
- en fonction des paramètres, la vue matérialisée sera tenue à jour ou non quand la table change
- Entretien des vues en temps réel : coûteux (et peu souvent mis en oeuvre)

À suivre :

- Requetes à ne pas écrire
- Data Warehouses