

bases de données
avancées
SASUGG2

Les Vues relationnelles

Présenté par:

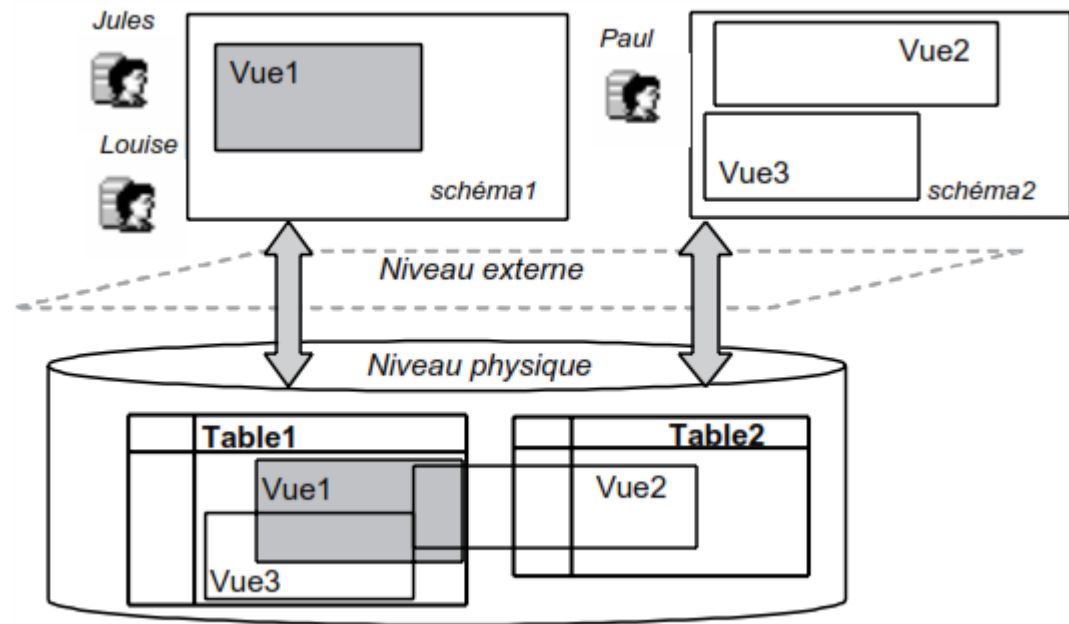
Amal HALFAOUI (Epse GHERNAOUT)

amal.halfaoui@univ-tlemcen.dz

amal.halfaoui@gmail.com

Les vues (Définition)

Est une vision logique des données contenues dans une ou plusieurs tables. Elle est considérée comme une table virtuelle qui n'a pas d'existence propre, aucune donnée ne lui est associée et sa description est stockée sous la forme d'une requête.



Les vues (Définition)

Une vue est créée à l'aide d'une instruction SELECT appelée « requête de définition ». Cette requête interroge une (ou plusieurs) table(s) ou vue(s). Une vue se recharge chaque fois qu'elle est interrogée.

Utilité

- **Confidentialité**

Les utilisateurs n'ont le droit d'accéder qu'aux vues (Exp: vue sur la table Employé destinée aux employés ne sélectionne pas la colonne Salaire contrairement à la vue destinée aux chefs de service.)

- **Cacher la complexité des données aux utilisateurs**

Vue simplifiée des données

Simplifier les requêtes des utilisateurs non spécialistes

- **Présenter différentes perspectives sur les données aux utilisateurs**

Renommage des colonnes

- **Requêtes complexes**

Qui ne peuvent être faites sans vues

Qui sont fréquemment réalisées

Syntaxe

```
CREATE [OR REPLACE] VIEW nom_vue [ ( nv_nom_col)* ] AS subquery  
[WITH CHECK OPTION [CONSTRAINT nom_contrainte] ]  
[WITH READ ONLY];
```

Exemple

```
CREATE OR REPLACE VIEW Petit_pays  
AS SELECT * FROM Pays WHERE surface < 100
```

Exemple

```
CREATE VIEW PilotesAF
AS SELECT *
FROM Pilote
WHERE compa = 'AF';
```

Pilote

brevet	nom	nbHVol	adresse	compa
PL-1	Soutou	890	Castanet	CAST
PL-2	Laroche	500	Montauban	CAST
PL-3	Lamothe	1200	Ramonville	AF
PL-4	Albaric	500	Vieille-Toulouse	AF
PL-5	Bidal	120	Paris	ASO
PL-6	Labat	120	Pau	ASO
PL-7	Tauzin	100	Bas-Mauco	ASO

```
CREATE VIEW Etat_civil
AS SELECT nom, nbHVol, adresse,
compa FROM Pilote;
```

Vue d'une vue

La requête de définition de la vue porte sur une autre vue.

CREATE OR REPLACE VIEW

EtatCivilPilotesAF **AS**

SELECT nomPil,heuresPil,adressePil

FROM PilotesAF

Interrogation d'une vue

Une fois créée, une vue s'interroge comme une table par tout utilisateur, sous réserve qu'il ait obtenu le privilège en lecture directement (GRANT SELECT ON nomVue TO...)

Somme des heures de vol des pilotes d'Air France.
`SELECT SUM(nbHVol) FROM PilotesAF;`

+-----+
SUM(nbHVol)
+-----+
1700.00
+-----+

Classification

Requête de définition	Vue simple	Vue complexe
Nombre de tables	1	1 ou plusieurs
Fonction	non	oui
Regroupement	non	oui
Mises à jour possibles ?	oui	pas toujours

Une vue monotable est définie par une requête SELECT ne comportant qu'une seule table dans sa clause FROM.

Vues modifiables

Lorsqu'il est possible d'exécuter des instructions INSERT, UPDATE ou DELETE sur une vue, cette dernière est dite « modifiable » (updatable view).

Pour qu'une vue simple soit modifiable, sa requête de définition doit respecter les critères suivants :

- pas de directive DISTINCT, de fonction (AVG, COUNT, MAX, MIN, SUM,...) d'expression dans le SELECT ;
- pas de GROUP BY, ORDER BY ou HAVING.

Vues modifiables

Les 3 opérations sont possibles

```
DELETE FROM piloteAF  
WHERE compa = 'ASO';
```

```
UPDATE Etat_civil  
SET nbHVol = nbHVol*2  
WHERE nom = 'Lamothe';
```

Insertion n'est pas conseillée quand le champs clé n'est pas renseigné sinon pas possible

```
INSERT INTO Etat_civil VALUES ('Raffarin',10,'Poitiers','ASO') ;
```

Ajout d'un pilote qui n'est pas de 'AF' !

```
INSERT INTO VALUES  
( 'PL-9','Caboche', 600,'Rennes','ASO');  
PilotesAF
```

Directive CHECK OPTION (constraints)

```
CREATE OR REPLACE VIEW PilotesAF  
AS SELECT * FROM pilote where comp='AF'  
WITH CHECK OPTION;
```

INSERT INTO PilotesAF VALUES ('PL-9','Caboche',600,'Rennes','ASO');
ERROR 1369 (HY000): CHECK OPTION failed 'bdsoutou.PilotesAF'

```
CREATE OR REPLACE VIEW Petit_pays  
AS SELECT * FROM Pays WHERE surface < 100  
WITH CHECK OPTION CONSTRAINT CK_Petit_pays;
```

Si :

INSERT INTO Petit_pays values ("Chine", "Pékin", 2000, 500) → Impossible!!! Car surface est égale à 500

Exemple

```
CREATE OR REPLACE VIEW Petit_pays  
AS SELECT * FROM Pays WHERE surface < 100  
WITH READ ONLY;  
INSERT INTO Petit_pays values ("Chine", "Pékin", 2000, 500):
```

Directive READ ONLY (contraints)

Aucune Opération n'est possible à travers la vue

```
CREATE OR REPLACE VIEW Petit_pays  
AS SELECT * FROM Pays WHERE surface < 100  
WITH READ ONLY;
```

Si :

INSERT INTO Petit_pays values ("Pays", « Ville", 2000, 90) → Impossible!!! Même si la surface est 100

Vues complexes

- Une vue complexe est caractérisée par le fait qu'elle contient, dans sa définition, **plusieurs tables (jointures)** et une **fonction** appliquée à des regroupements ou à des expressions. La mise à jour de telles vues n'est pas toujours possible.

Pour pouvoir modifier une vue complexe, les restrictions sont les suivantes :

- La requête de définition ne doit pas contenir de sous-interrogation (jointure procédurale).
- Il n'est pas possible d'utiliser d'opérateur ensembliste (sauf UNION [ALL]).

Vues complexes (multitable)

```
CREATE VIEW Pilotes_multi_AF
AS SELECT p.brevet, p.nom,
        p.nbHVol, c.ville, c.nomComp
FROM Pilote p, Compagnie c
WHERE p.compa = c.comp AND p.compa = 'AF';
```

Compagnie

comp	nrue	rue	ville	nomComp
AF	124	Port Royal	Paris	Air France
SING	7	Camparols	Singapour	Singapore AL

Pilote

brevet	nom	nbHVol	compa
PL-1	Louise Ente	450	AF
PL-2	Jules Ente	900	AF
PL-3	Paul Soutou	1000	SING

```
CREATE VIEW Moyenne_Heures_Pil
AS SELECT compa,
        AVG(nbHVol) moyenne
FROM Pilote GROUP BY compa;
```

compa	moyenne
AF	675
SING	1000

Confidentialité

La confidentialité est une des vocations premières des vues. Outre l'utilisation de variables d'environnement, il est possible de restreindre l'accès à des tables en fonction de moments précis.

```
CREATE VIEW VueDesCompagniesJoursFeries  
AS SELECT * FROM Compagnie  
WHERE DATE_FORMAT(SYSDATE(), '%W')  
IN ('Sunday', 'Saturday')
```

Restriction en lecture de la table Compagnie, les samedi et dimanches. Mises à jour possibles à tout moment.


```
CREATE VIEW VueDesPilotesJoursOuvrables  
AS SELECT * FROM Pilote  
WHERE  
CURTIME()+0 BETWEEN 83000 AND 173000  
AND DATE_FORMAT(SYSDATE(), '%W')  
NOT IN ('Sunday', 'Saturday')  
WITH CHECK OPTION;
```

Restriction, en lecture et en écriture (à cause de WITH CHECK OPTION), de la table Pilote les jours ouvrables de 8h30 à 17h30.

Attribution des droits

```
Create view Information_me_concernant  
As select *  
From employe  
Where upper(nom) = upper (user)
```

Chaque utilisateur peut
visualiser un seul tuple de la
relation: celui qui le
concerne



```
GRANT SELECT ON Information_me_concernant  
TO public
```

```
GRANT insert ON VueDesCompagniesJoursFeries  
TO 'Paul'@'localhost';
```

Déclencheurs sur les vues

- **Déclencheur `INSTEAD OF`**

Un déclencheur `INSTEAD OF` permet de mettre à jour une vue multitable qui ne pouvait être modifiée directement par `INSERT`, `UPDATE` ou `DELETE`. *le déclencheur programmera des actions au lieu d'insérer, de modifier ou de supprimer une vue.*

Caractéristiques

Les déclencheurs `INSTEAD OF` :

- font intervenir la clause `FOR EACH ROW` ;
- ne s'utilisent que sur des vues ;
- ne font pas intervenir les options `BEFORE` et `AFTER`.
- Il n'est pas possible de spécifier une liste de colonnes dans un déclencheur `INSTEAD OF UPDATE` , le déclencheur s'exécutera quelle que soit la colonne modifiée.
- Il n'est pas possible d'utiliser la clause `WHEN` dans un déclencheur `INSTEAD OF`.

Déclencheurs sur les vues

Pilote

brevet	nom	nbHVol	compa
PL-1	Agnès Bidal	450	AF
PL-2	Aurélia Ente	900	AF
PL-3	Florence Périssel	1000	SING

Compagnie

comp	nrue	rue	ville	nomComp
AF	124	Port Royal	Paris	Air France
SING	7	Camparols	Singapour	Singapore AL



```
CREATE VIEW VueMultiCompPil
AS SELECT c.comp, c.nomComp, p.brevet, p.nom, p.nbHVol
FROM Pilote p, Compagnie c
WHERE p.compa = c.comp;
```

COMP	NOMCOMP	BREVET	NOM	NBHVOL
AF	Air France	PL-1	Agnès Bidal	450
AF	Air France	PL-2	Aurélia Ente	900
SING	Singapore AL	PL-3	Florence Périssel	1000

Déclencheurs sur les vues

Code PL/SQL	Commentaires
<pre>CREATE TRIGGER TrigAulieuInsererVue INSTEAD OF INSERT ON VueMultiCompPil FOR EACH ROW DECLARE v_comp NUMBER := 0; v_pil NUMBER := 0;</pre>	Déclaration de la substitution de l'événement déclencheur.
<pre>BEGIN SELECT COUNT(*) INTO v_pil FROM Pilote WHERE brevet = :NEW.brevet; SELECT COUNT(*) INTO v_comp FROM Compagnie WHERE comp = :NEW.comp; IF v_pil > 0 AND v_comp > 0 THEN RAISE_APPLICATION_ERROR(-20102, 'Le pilote et la compagnie existent déjà!'); ELSE IF v_comp = 0 THEN INSERT INTO Compagnie VALUES (:NEW.comp, NULL, NULL, NULL, :NEW.nomComp); END IF; IF v_pil = 0 THEN INSERT INTO Pilote VALUES (:NEW.brevet, :NEW.nom, :NEW.nbHVol, :NEW.comp); END IF; END IF; END;</pre>	Corps du déclencheur.
	Cas d'erreur.
	Ajout dans la table Compagnie.
	Ajout dans la table Pilote.
<pre>/</pre>	

Référence

- Livre SQL pour Oracle, de C.Soutou.