SQL : Les vues

Afpa - Stéphane PONTONNIER 2022

Plan

- Définitions
- Création
- Suppression
- Propriétés
- Manipulation des données d’une vue
- Options d’une vue

Définition

SQL

Vue

- Comme une table virtuelle : physiquement inexistante
- Définie par un nom et une requête SQL
- Pouvant être interrogée comme une table
- Pouvant, sous certaines conditions, être mise à jour

Remarque

Une mise à jour des tables à partir desquelles une vue est définie se propage aux vues

Définition

SQL

Avantages

- Rassembler les données logiques même si elles sont définies sur plusieurs tables
- Simplifier les requêtes en masquant la complexité du schéma
- Sécuriser l'accès à certaines colonnes si on ne les sélectionne pas avec la requête de la vue

Inconvénients

- Restriction sur les mises à jour
- Coût : chaque appel d'une vue vaut l'exécution de la requête SELECT précisée à la création

Création

SQL

Syntaxe de création

CREATE [OR REPLACE] VIEW nom_vue [(colonnes)] **AS**
requêteSelect
[WIDTH [CASCADED | LOCAL] CHECK OPTION]

Exemple

CREATE OR REPLACE VIEW marseillais **AS**
SELECT num, nom, prenom
FROM personne
WHERE ville=”MARSEILLE”;

Une vue peut être interrogée comme une table

SELECT * **FROM** marseillais;

Suppression

SQL

Syntaxe de suppression
DROP VIEW nom_vue;

Propriétés

SQL

On peut créer une vue multi-tables

```
CREATE OR REPLACE VIEW vehicule_marseillais AS
SELECT marque, modele, ville, nom, prenom
FROM personne, vehicule
WHERE ville=”MARSEILLE”
AND nump=num;
```

Cette vue peut aussi être interrogée comme une table

```
SELECT * FROM vehicule_marseillais;
```

Propriétés

SQL

On peut créer une vue à partir d'une autre vue

```
CREATE OR REPLACE VIEW cadre_marseillais AS
SELECT *
FROM marseillais
WHERE salaire >=2000;
```

Cette vue peut aussi être interrogée comme une table

```
SELECT * FROM cadre_marseillais;
```

Propriétés

SQL

Exercice 1

En utilisant les vues, sélectionnez les personnes qui ont le plus grand nombre de voitures en utilisant les deux fonctions d'agrégation ***max*** et ***count***.

Manipulation des données d'une vue

SQL

Conditions pour la mise à jour

- Si la vue est définie à partir de plusieurs tables, il faut que les modifications concernent une seule table.
- Il ne faut pas que la requête de la vue contienne les mots-clés suivants :
  DISTINCT
  LIMIT
  GROUP BY
  HAVING
  UNION
  Une fonction d'agrégation

Manipulation des données d'une vue

SQL

Modifier le modèle est possible

```
UPDATE vehicule_marseillais
SET modele=”Focus” WHERE marque =”Ford”;
```

Modifier le nom est possible

```
UPDATE vehicule_marseillais
SET nom=”Mercure” WHERE prenom =”Sophie”;
```

Modifier le modèle et le nom est impossible

```
UPDATE vehicule_marseillais
SET modele=”Focus”,
SET nom=”Mercure”
WHERE prenom =”Sophie”;
```

Manipulation des données d'une vue

SQL

Conditions pour l'insertion = conditions pour la maj +

- Les colonnes *not null* d'une table, n'ayant pas une valeur par défaut, doivent être présentes dans le **SELECT** de création de la vue
- Il ne faut pas avoir de colonnes dupliquées dans la vue (**SELECT *** dans une jointure)

Manipulation des données d'une vue

SQL

Insérer le tuple suivant est possible

```
INSERT INTO vehicule_marseillais
SET modele=”Ibiza” WHERE marque =”Seat”;
```

Insérer le tuple un tuple dans une vue qui implique deux insertions dans deux tables différentes (un premier dans la table véhicule et un deuxième dans la table personne) est impossible

```
INSERT INTO vehicule_marseillais
SET nom=”Green”,
prenom=”Mickael”,
modele=”Ibiza”,
 WHERE marque =”Seat”;
```

Manipulation des données d’une vue

SQL

Conditions pour la suppression = conditions pour la maj +

- La vue est mono-table

Options d'une vue

SQL

Syntaxe de création
CREATE [OR REPLACE] VIEW nom_vue [(colonnes)] **AS**
requêteSelect
[WITH [CASCADED | LOCAL] CHECK OPTION]

Explications

- *check option* permet de modifier une vue à condition que le tuple modifié appartienne encore à la vue après la mise à jour
- *check option* interdit également l'insertion dans une vue d'un tuple contenant une valeur qui ne satisfait pas la condition de sélection de la vue.
- *WITH LOCAL CHECK OPTION* : on vérifie seulement les conditions de la vue.
- *WITH CASCADED CHECK OPTION* (par défaut) : on vérifie les conditions de la vue et celles des vues à partir desquelles on l'a créée.

Options d'une vue

SQL

Considérons la vue suivante

```
CREATE OR REPLACE VIEW cadre_marseillais AS
SELECT *
FROM marseillais
WHERE salaire >= 2000
WIDTH CHECK OPTION;
```

Il est possible d'ajouter un tuple à cette vue

```
INSERT INTO cadre_marseillais VALUES(8, 'Freeman', 'Morgan', 2500);
```

Ajouter un tuple qui ne remplit pas les conditions de création d'une vue est impossible

```
INSERT INTO cadre_marseillais VALUES(9, 'Kaho', 'Rebecca', 2500);
```