|  |  |
| --- | --- |
|  |  |
| **SUPPORT DE COURS SLAM1** | |

**SQL**

|  |  |  |
| --- | --- | --- |
| **date** | **révision** | |
| 2019 | Timothée Robert : création du document | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  | |  |

|  |  |
| --- | --- |
| Les Vues - introduction |  |

## Sources

SQL Server 2014 : F. Brouard & alii, Eyrolles

Discovering SQL: A. Kriegel, Wrox (Wiley)

Beginning SQL : P. Wilson & alii, Wrox (Wiley)

## Objectif

Nous allons dans ce cours et les suivants aborder la confidentialité et la sécurité des données hébergées dans une base SQL.

Les vues sont un moyen d’assurer la confidentialité des données : elles agissent comme des « fenêtres » sur la base de données.

Une bonne pratique de développement est de n’accéder qu’à des vues et non pas des tables.

## Généralités

Seules les tables contiennent des données. Mais pour l’utilisateur, une vue apparaît comme une table et permet, en plus du contrôle d’accès aux données :

* Masquer la complexité un modèle relationnel
* Simplifier les requêtes complexes

Une Vue est considérée comme une table virtuelle : elle ne nécessite pas d’allocation mémoire pour contenir les données.

Seule sa structure est stockée dans le dictionnaire des données de la base de données.

Elle est créé au moyen d’une requête SELECT, appelée requête de définition.

## Syntaxe d’une vue

CREATE VIEW [*nom\_schema*] *nom\_vue* [(colonne1 [,…])]

AS requete\_SELECT

;

GO

Entre [] des éléments optionnels.

Remarque :

* on ne peut pas mettre d’instruction ORDER BY dans la requête de définition d’une vue
* si aucun nom de colonne n’est spécifié, les colonnes prennent le nom des colonnes retourné par l’instruction SELECT

## Classification des vues



Source : Brouard : SQL Server 2014

# Données utilisée pour les exemples

## Modèle relationnel

**T\_compagnie\_cmp** (cmp\_comp, cmp\_pays, cmp\_nom, pil\_chef)

Clé primaire : cmp\_comp

**T\_pilote\_pil** (pil\_brevet, pil\_prenom, pil\_nompil\_embauche, pil\_fct, pil\_salaire, cmp\_comp, pil\_responsable)

Clé primaire : avi\_immat

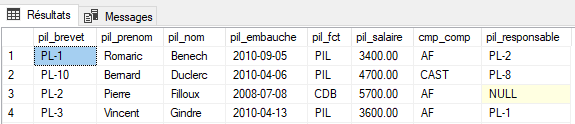
Clé étrangères:

cmp\_comp reference T\_compagnie\_cmp(cmp\_comp)

pil\_responsable reference T\_pilote\_pil(pil\_brevet)

## Tables de données

Extrait table VUES.T\_Pilote\_Pil



Extrait table VUES.T\_compagnie\_cmp



## Emplacement

Sur SQL Server les vues sont stockées dans le classeur Vues au niveau de l’explorateur d’objets :

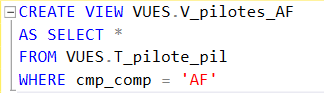


# Vues simples

## Exemples

Soit les 2 vues suivantes, toutes les 2 constituées à partir de la table VUES.T\_Pilote\_pil

1ère vue : Vue horizontale décrivant les pilotes d’Air France



2ème vue : Vue verticale constituée de certaines colonnes uniquement

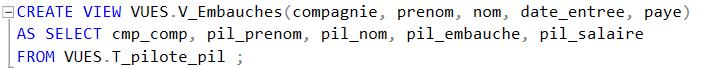
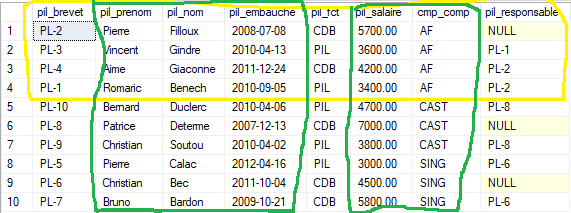


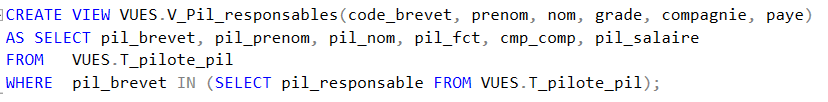
Table T\_pilote\_pil : en jaune les données correspondant à la 1ère vue, en vert les données correspondant à la 2ème vue.



## Combiner une restriction avec un choix de colonne

On peut très bien dans une vue mixer des éléments de l’instruction SELECT et de la restriction WHERE pour filtrer les données à la fois sur les lignes et les colonnes.

Exemple : extraction des pilotes qui sont déclarés en tant que responsable pour la compagnie



## Interrogation d’une vue simple

Une vue s’utilise comme une table.

Cf exercices TP associé.

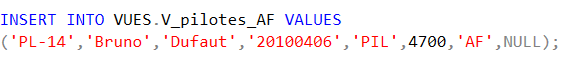
## Vue modifiable : Mises à jour au travers d’une Vue simple

### MAJ Possible sous conditions

Les modifications des données sous-jacentes sont possibles sous réserve

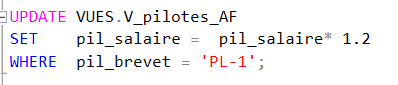
1. Que les données insérées disposent de toutes les colonnes nécessaires (non NULLable) de la table insérée
2. De respecter les contraintes d’intégrité

Exemple 1 : ajout d’un pilote au travers de la vue V\_pilotes\_AF



L’insertion dans la table sous jacente Vues.T\_pilote\_pil au travers de la Vue Vues.V\_pilotes\_AF est possible car les 2 conditions sont vérifiées.

Exemple 2 : modification d’un pilote



La colonne pil\_salaire de la table Vues.T\_pilote\_pil est modifiée pour le pilote ‘PL-1’ au travers de la vue.

### MAJ impossible

Les MAJ sont impossibles lorsqu’une des 2 conditions n’est pas remplie.

Exemple : tentative d’ajout à la table VUES.T\_pilote\_pil au travers de la vue VUES.V\_Embauches



Ce code génère l’erreur suivante :



# Vues complexes

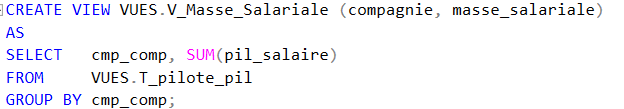
## Introduction

Une vue complexe est généralement caractérisée par le fait de manipuler plusieurs tables (jointures) ou alors d’effectuer un agrégat (fonctions d’agrégation).

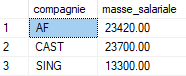
## Exemples

Exemple 1 : Vue définissant un agrégat

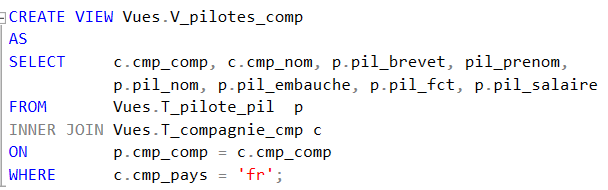
La vue suivante donne la masse salariale par compagnie :



Résultat :



Exemple 2 : Vue issue d’une jointure



Résultat d’une sélection triée par compagnie, nom et prénom du pilote



## Vue modifiable : Mises à jour au travers d’une Vue complexe

En règle générale, il n’est pas possible d’avoir une Vue complexe modifiable , sauf cas particulier. En effet l’identification des lignes de base de la table sous-jacente qu’on souhaite modifier (ou ajouter) est perdue.

Mais dans les cas particuliers où on dispose de la clef primaire de la table qu’on souhaite modifier à travers la vue, la MAJ est possible.

Exemple de cas de MAJ possible à travers une Vue complexe :

Pour la vue précédente, V\_pilotes\_comp, l’insertion de donnée n’est pas possible, car la modification porterait sur plusieurs tables, en revanche la MAJ du salaire ou un changement de nom de la compagnie sont possibles.

# Synthèse et compléments

## Synthèse

L’intérêt des vues est multiple :

* Sécurité des accès avec filtres en lignes (WHERE) et colonnes (clause SELECT)
* Simplification et synthétisation: on peut clarifier la vision des données
* MAJ : la MAJ est généralement possible pour les Vues simples et possible sous certaines conditions avec les vues complexes (de type jointure)

Il est considéré comme une bonne pratique qu’une application développée utilise des Vues plutôt que les tables d’une BDD

## Compléments

Hors périmètre du SIO, il y a beaucoup d’autres types de Vues : Vues indexées, Vues paramétrées, Vues Matérialisées (sous Oracle).