



Les Schémas

SQL serveur 2005

Philippe GARRAUD
lundi 17 septembre 2007

Objectifs

Savoir utiliser les schémas sous sql serveur 2005

| | |
|---|----------|
| 1. PRESENTATION DES SCHEMAS..... | 3 |
| 1.1. Objectifs et philosophie des schémas | 3 |
| 1.2. Méta modèle des schémas..... | 4 |
| 2. UTILISATION SIMPLIFIEE | 6 |
| 3. EXERCICE DE DECOUVERTE..... | 7 |
| 3.1. Créer la base..... | 7 |
| 3.2. Exploiter la base..... | 9 |
| 3.2.1. Consulter vol..... | 9 |
| 3.2.2. Planifier vol | 9 |
| 3.2.3. Planifier les ressources | 9 |

1. PRESENTATION DES SCHEMAS.

Les schémas ont été introduits par Microsoft avec la version 2005.

1.1.Objectifs et philosophie des schémas

Parmi les utilisateurs d'une base de données, il faut distinguer :

- Le propriétaire (dbo) data base owner, il a créé la base, il possède tous les droits. En général il est responsable de la création des tables.
- Les programmeurs : Leur mission principale est de programmer les procédures stockées, ils peuvent également créer certaines tables.
- Les exploitants, ils vont exécuter des requêtes et des procédures stockées.

En ce qui concerne les programmeurs ils créent des objets, procédures stockées et tables et les rangent dans un ou plusieurs schémas. La base se trouve donc partitionnée en plusieurs schémas.

Typiquement : les programmeurs auront donc des droits de création de tables sur les schémas, les exploitants auront des droits d'accès aux données des tables des schémas.

Les utilisateurs peuvent être regroupés en rôles.

- Souvent les rôles correspondent à des catégories d'exploitants : comptable, commerciaux etc, exécutant les mêmes applications.
- Les utilisateurs sont en général des personnes physiques.

Il est souvent intéressant de gérer les droits au niveau des rôles, ce qui évite de devoir les modifier à chaque embauche de personnel.

Par la suite nous parlerons toujours d'utilisateur, englobant également dans ce concept les rôles.

1.2.Méta modèle des schémas.

On appelle ici objet : un type, une table, une procédure stockée

Un objet appartient à un et un seul schéma, par défaut il appartient au schéma `dbo`.
Pour ajouter une table à un schéma, il suffit d'écrire :

```
Create table MonSchema.Matable  
(  
)
```

Pour avoir le droit de créer un objet dans un schéma, il faut en être propriétaire ou au moins posséder le droit `alter`.

Pour déplacer un objet d'un schéma à un autre vous pouvez utiliser :

```
Alter schema MonSchema transfer UneAutreTable.
```

Il faut être posséder le droit `control` sur `UneAutreTable`, et le droit `alter` sur le schéma cible.

Un schéma peut contenir plusieurs objets.

Un schéma appartient à un et un seul utilisateur, seul le propriétaire ou un utilisateur possédant le droit `alter` peut créer ou déplacer des objets vers ce schéma. Le propriétaire d'un schéma a tous les droits sur les objets du schéma.

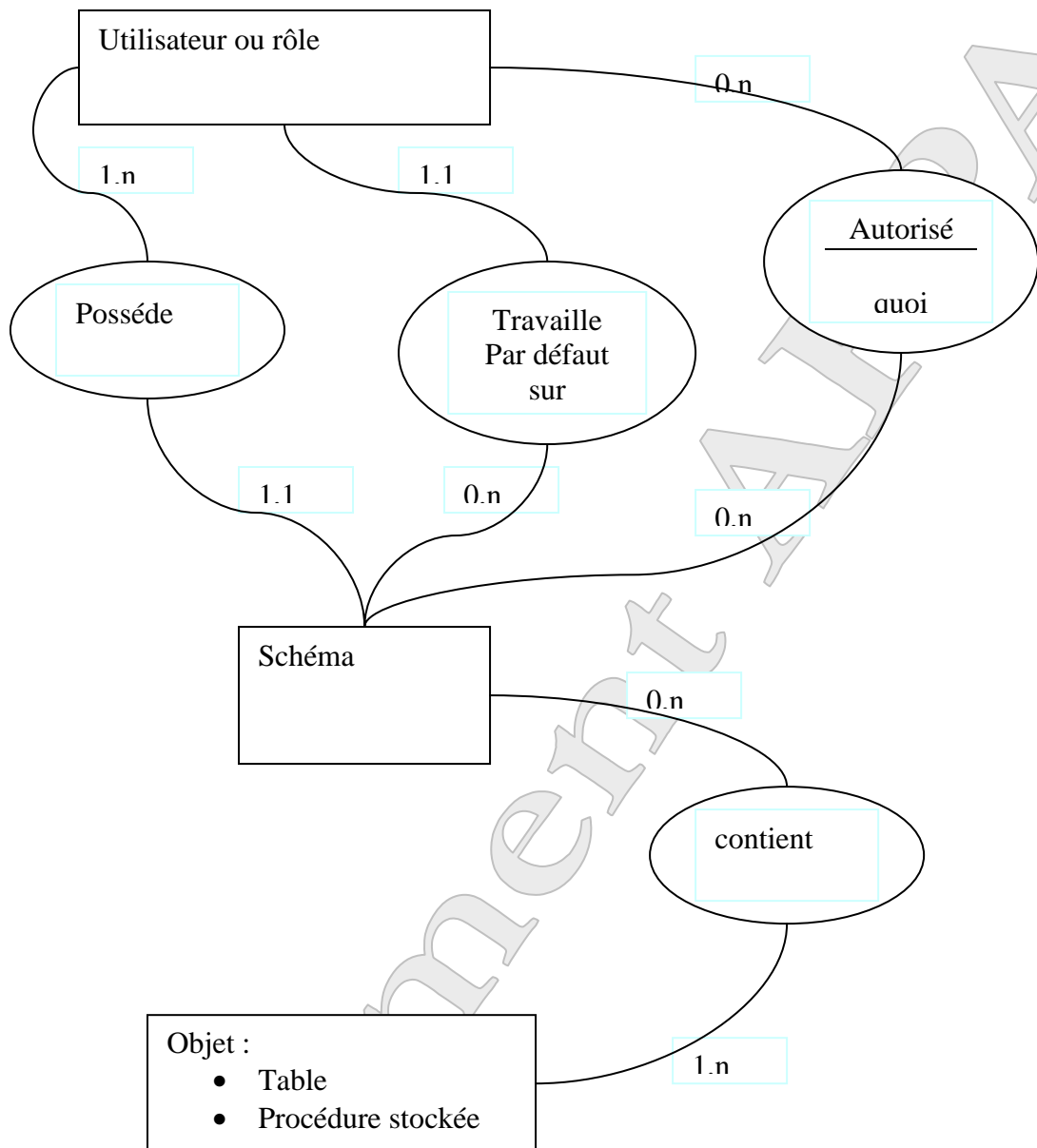
Un utilisateur peut posséder plusieurs schémas, mais il en a un par défaut.

Les utilisateurs peuvent avoir des droits sur des schémas, par exemple le droit `select`, implique un droit `select` sur toutes les tables du schéma. Un schéma peut à l'inverse accorder des droits à plusieurs utilisateurs. Pour gérer ces droits il est souvent nécessaire d'être `dbo`.

`dbo` voit tous les schémas.

Noter bien que le schéma complet d'une table est
`NomServeur.NomBase.NomSchéma.NomTable`

Le méta modèle est dessiné ci-dessous dans le formalisme MCD :



2. UTILISATION SIMPLIFIEE

Par défaut tout objet est crée dans le schéma dbo.

Par défaut dbo est propriétaire du schéma dbo.

Le schéma par défaut d'un utilisateur est dbo.

Il est raisonnable de créer toutes les tables dans le schéma dbo et avec l'utilisateur dbo.

Les procédures stockées pourront simplement être créées également dans le schéma dbo ou dans des schémas spécifiques. Dans les cas simples on peut ne travailler qu'avec le seul schéma dbo.

Les droits d'accès des exploitants peuvent toujours être gérés au niveau objet. En ce cas il ne faut leur donner aucun droit sur les schémas.

3. EXERCICE DE DECOUVERTE

Toutes les manipulations ci-dessous peuvent s'effectuer par des lignes de commandes sql ou avec l'outil graphique d'administration.

3.1. Créer la base

Nous allons reprendre notre base Air régis et la répartir sur plusieurs schémas :

Ignorer les schémas et les rôles préétablis, remarquer simplement la présence du schéma par défaut dbo.

Pour commencer nous serons logués dbo

Créons deux utilisateurs :

- un programmeur P1 responsable du développement en ce qui concerne les avions et les lignes.
- un programmeur P2 responsable du développement de la planification.

Créons trois schémas. :

- Schéma Avions avec les tables Avions, Types, Constructeurs
- Schéma Lignes avec les tables Aéroports et Vols
- Schéma Plan avec les tables pilotes et Planification.

Nous pouvons commencer par créer les utilisateurs.

Puis nous créons les schémas vides dans un premier temps.

Définissons P1 comme propriétaire des Schémas Avions et Lignes.

Définissons P2 comme propriétaire du schéma Plan.

Affectons par défaut le schéma Avions à P1, et le schéma Plan à P2.

Autorisons Create table au niveau des autorisations de la base à P1 et P2.

Logons nous maintenant P1 :

Nous pouvons créer dans le schéma Avion les tables Constructeur, type, avions ;

Puis dans le schéma Lignes les tables Aéroports et Vols.

Prendre garde à créer les tables dans les bons schémas en utilisant la syntaxe :

```
Create table MonSchema.Matable  
(  
)
```

Nous n'avons pas de problèmes pour créer les contraintes d'intégrités, puisque P1 est propriétaires des deux schémas.

Logons nous maintenant P2 :

Nous pouvons créer les tables Pilotes et Planification.

Remarquer que nous ne voyons pas les tables Avions, Aéroport etc.

Nous ne pouvons donc pas créer la contrainte d'intégrité entre Planification et Avions.

Logons nous maintenant dbo :

Nous voyons toutes les tables, nous pouvons donc compléter les contraintes d'intégrité entre schéma.

Remarque de méthode : Séparer une base de données en plusieurs schémas n'est pertinent que si cette base se décompose bien en deux ensembles quasiment disjoints.

3.2.Exploiter la base.

3.2.1. Consulter vol

Sous dbo créons un utilisateur visiteur capable de lire les informations sur les lignes.
Accordons lui donc le droit select sur le schéma ligne

Grant select on ligne to visiteur

Logons nous visiteur :

Nous pouvons effectivement afficher le contenu des tables Aéroports et Vols

3.2.2. Planifier vol

Sous dbo créons un utilisateur PlanVol capable de remplir la table vol.
Accordons lui les droits insert, update et delete sur le schéma ligne.

Logons nous PlanVol

Nous pouvons effectivement remplir la table Vols

3.2.3. Planifier les ressources

Sous dbo créons un utilisateur PlanRessources capable de remplir la table planification
Accordons lui les droits insert, update et delete sur le schéma plan.

Logons nous PlanRessource

Nous pouvons effectivement remplir la table planification