

SQL pour les NULS

PAR DES NULS

Damien GIDON & Thibault CAM

Enoncé

L'objectif de cet exercice est d'implémenter un petit modèle de gestion d'une école disons IN'TECH avec ses salles, professeurs et élèves, spécifié sous la forme de tests unitaires. Autrement dit : l'objectif de l'exercice est de faire passer en vert un jeu de tests.

Vous disposez pour cela d'une solution comprenant 3 projets :

- ITI.SchoolDatabase.Tests, contient les tests unitaires
- ITI.SchoolDatabase.Model, un modèle à implémenter, constitué exclusivement d'abstractions
- ITI.SchoolDatabase.Impl, une implémentation minimale du modèle (le code écrit ne permet rien de plus que de pouvoir compiler sans erreur) et qui contient un dossier SqlScript dans lequel se trouvent les scripts sql à remplir.

Pour faire tourner les tests unitaires il vous suffit de configurer le projet ITI.SchoolDatabase.Tests en tant que projet de démarrage puis d'exécuter la solution.

Pour les autres tests, à vous de faire en sorte qu'ils passent en vert. Pour cela, vous avez le droit de faire ce que bon vous semble dans le projet ITI.SchoolDatabase (modifier l'implémentation des méthodes, des propriétés ou des constructeurs, ajouter des membres *internal* ou *private*, créer de nouveaux types etc.) En revanche, vous ne devez pas modifier le projet ITI.SchoolDatabase.Tests.

Les tests unitaires sont là pour spécifier de façon détaillée les fonctionnalités attendues. Cependant voici une courte description de ce qui est attendu :

- On peut créer un élève, un professeur et une salle de classe.
 - Les salles de classe ont un format de nom bien précis (du type "E01").
 - Les salles de classe ont une capacité > 0
 - On peut affecter un professeur principal à un élève

▲ **N'oubliez pas de modifier les 2 ConnectionStrings dans le appsettings.json dans ITI.SchoolDatabase.Impl** ▲

- Initialisation de la Base de données

- Créer une base de données locale qui se nomme "itiSchoolDB"
- Créer la table Teacher

	Column Name	Data Type	Allow Nulls
🔑	Guid	uniqueidentifier	<input type="checkbox"/>
	Name	nvarchar(150)	<input type="checkbox"/>
	Course	nvarchar(100)	<input type="checkbox"/>
	Orientation	nvarchar(2)	<input checked="" type="checkbox"/>
	BirthDate	date	<input type="checkbox"/>
	IsInternal	bit	<input type="checkbox"/>

- Créer la table Classroom

	Column Name	Data Type	Allow Nulls
🔑	Guid	uniqueidentifier	<input type="checkbox"/>
	Name	nvarchar(4)	<input type="checkbox"/>
	Capacity	int	<input type="checkbox"/>
	Projector	bit	<input type="checkbox"/>
	Teacher	uniqueidentifier	<input checked="" type="checkbox"/>

- Créer la table Student

	Column Name	Data Type	Allow Nulls
🔑	Guid	uniqueidentifier	<input type="checkbox"/>
	Name	nvarchar(150)	<input type="checkbox"/>
	Semestre	int	<input type="checkbox"/>
	Orientation	nvarchar(2)	<input checked="" type="checkbox"/>
	BirthDate	date	<input type="checkbox"/>
	MainTeacher	uniqueidentifier	<input checked="" type="checkbox"/>

- Remplir ces 3 tables à l'aide des Script dans => ITI.SchoolDatabase.Impl
SqlScript/Insert

-

▲ **Ne pas modifier ces scripts « INSERT » sinon vos tests ne passeront plus** ▲

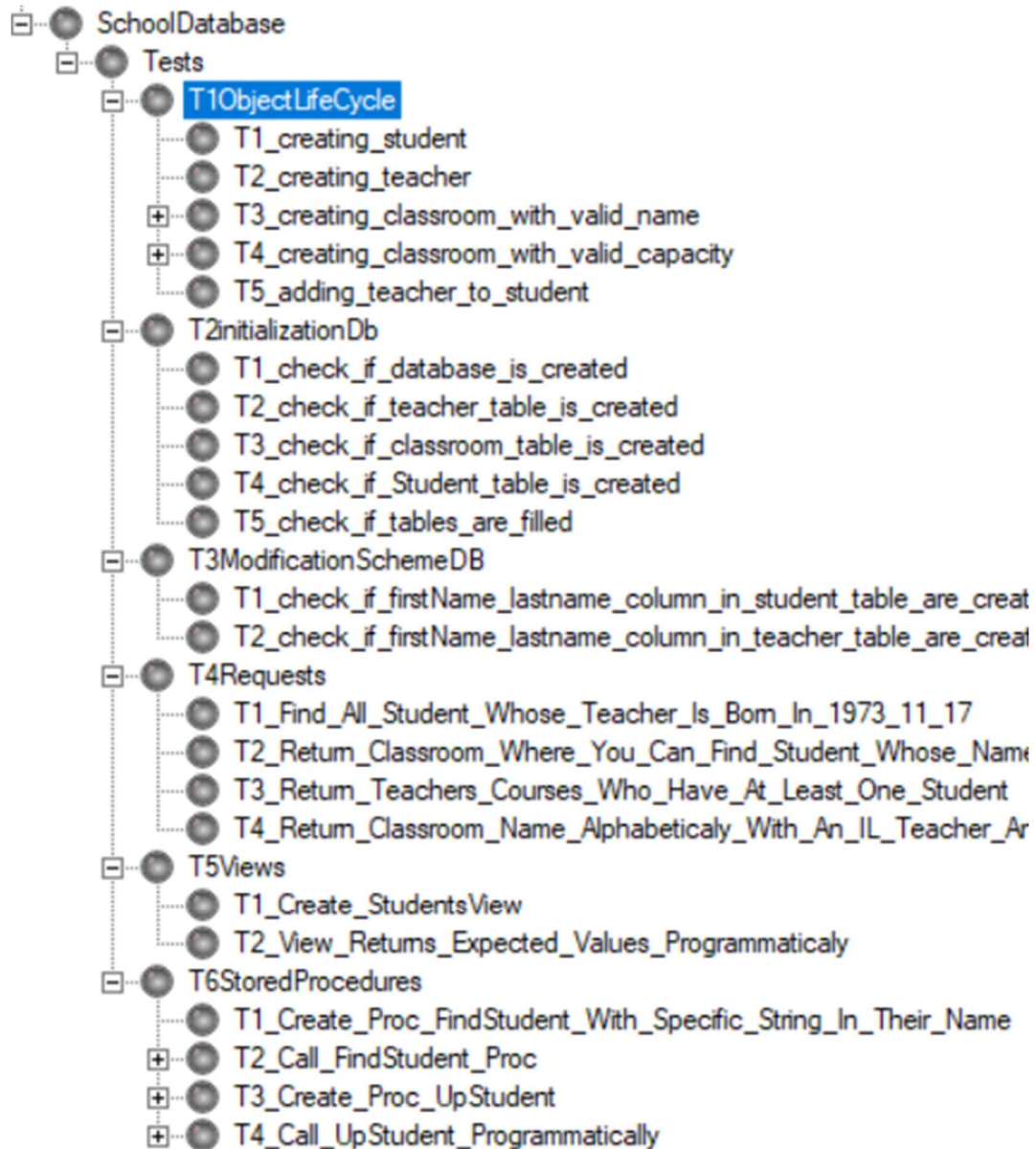
▲ **Afin de séparer les requêtes du fichier Requests pour le T4, des « GO » ont été placés dans certain fichiers afin de pouvoir les tester séparément, veuillez les respecter et ne pas les supprimer** ▲

Le T5_check_if_tables_are_filled du T2InitializationDb sera **VERT** si vous n'avez rien changé

- Modifier la Table Student & Teacher (A faire dans SqlScript/ModificationScheme)
 - Créer 2 nouvelles colonnes FirstName et LastName et les remplir avec les données de la colonne Name
 - Idem pour Teacher mais uniquement pour les professeurs nés après 1975

- Requête de base. (A faire dans SqlScript/Requests)
 - Retourner tous les étudiants qui ont un professeur né le 17/11/1973
 - Retourner les salles où l'on peut trouver des étudiants dont le nom commence par "Thibau" quand ils sont avec leur enseignant principal
 - Retourner le cours des professeurs qui ont au moins un élève
 - Retourner les salles de classe par ordre alphabétique qui possède 1 projecteur et un professeur en orientation "IL"
- Création et utilisation de View SQL. (A faire dans SqlScript/View/View.sql)
 - Créer une View appelé StudentsView qui retourne la table student
 - Ecrire une méthode qui appelle la Vue et retourne les éléments
- Création et utilisation de procédure stockée. (A faire dans SqlScript/StoredProcedures)
 - Dans FindStudent, écrire une procédure stockée qui prend en paramètre une String et qui permet de retrouver tous les students qui possède cette string dans leur nom.
 - Ecrire une méthode qui appelle la Proc FindStudent et retourne les éléments
 - Dans UpStudent, Créer une procédure stockée qui prend une liste de Student en paramètre et qui va incrémenter le semestre de tous les élèves qui ne sont pas dans cette liste. Elle retournera les élèves qui ont dépassé le S10 avant de les deletes
 - Ecrire une méthode qui appelle la Proc UpStudent et retourne les éléments

Il y a 22 tests à passer au vert :



La partie 1 est indépendante du reste, de plus, les parties 4, 5 et 6 peuvent être faites dans l'ordre voulu du moment que la partie 2 est réalisée

Au boulot !