

The SQL Server 2-D Matrix Builder

Première partie : *comment utiliser*

« *The SQL Server 2-D Matrix Builder* »

Par Pascal Belaud, Microsoft France

<http://blogs.msdn.com/Pascal>

Janvier 2008

Sommaire

Pourquoi a-t-on besoin d'une matrice dans Microsoft SQL Server ?.....	1
Mode de licence de « The SQL Server 2-D Matrix Builder »	2
Installation de « The SQL Server 2-D Matrix Builder ».....	3
Vue d'ensemble de l'utilisation de « The SQL Server 2-D Matrix Builder ».....	6
Exemple d'utilisation n°1 : cas simple.....	11
Exemple d'utilisation n°2 : cas complexe.....	16
Exemple d'utilisation n°3 : cas encore plus complexe	24
Ensuite ?	36

Pourquoi a-t-on besoin d'une matrice dans Microsoft SQL Server ?

Le projet *The SQL Server 2-D Matrix Builder* est né d'un besoin que j'ai eu lors du développement de la plateforme de Reporting utilisée dans le projet *Vinci*, projet qui a pour vocation de gérer tout le backend nécessaire à l'organisation de l'événement *Microsoft TechDays 2008* :

<http://blogs.msdn.com/pascal/archive/tags/Vinci/default.aspx>

Nous avons eu besoin à un moment donné de fournir l'agenda complet des TechDays 2008. Pour cela, nous souhaitions que la matrice qui soit renvoyée contienne :

- En colonne, la liste des timeslots disponibles (11H-12H, 13H-14H, 15H-16H...)
- En ligne, la liste des salles disponibles (Salle 242, Salle 243, Amphi Bleu...)
- A l'intersection des deux, le code de la session technique qui est joué (DEV302, DAT402...)

Or la difficulté de générer une telle requête vient du fait que la liste des colonnes à renvoyer est complètement dynamique (la liste des salles utilisées peut varier en fonction des situations). Et c'est là

que m'est venu l'idée de tirer profit de la puissance de l'intégration de .NET dans le moteur de base de données *SQL Server 2005*, appelée aussi *SQLCLR*.

Pour faire simple, la Figure 1 expose un schéma de tables assez classique.

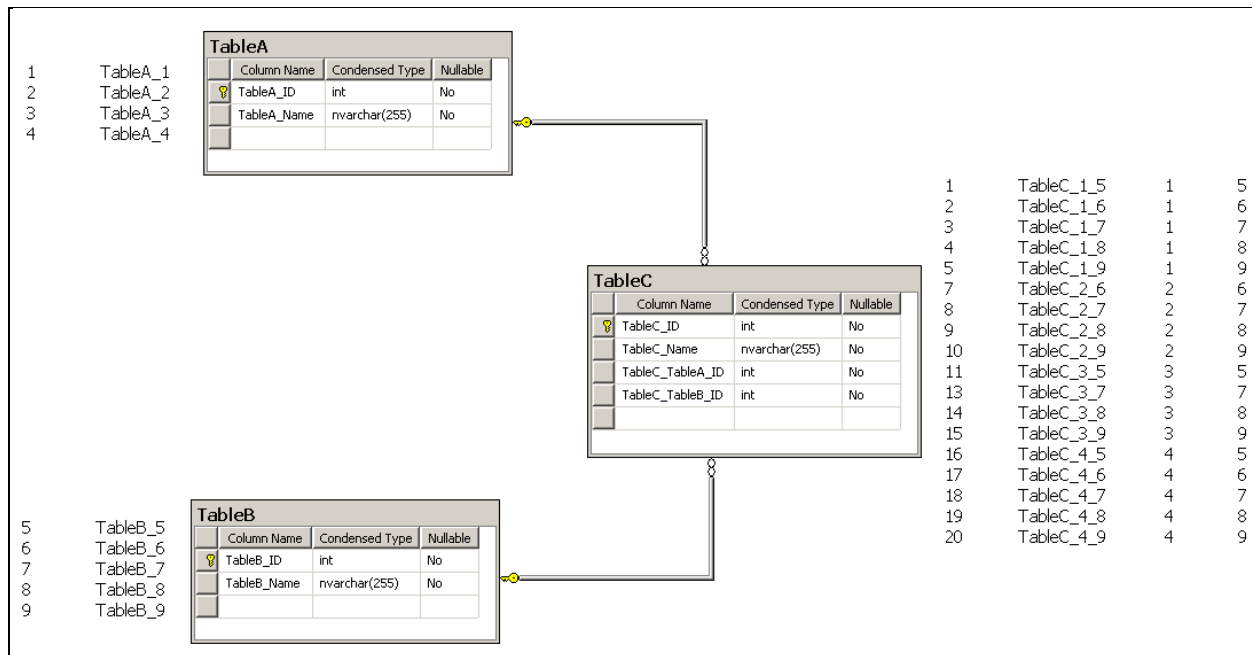


Figure 1

Ce que nous souhaitons, c'est pouvoir appeler une procédure stockée (ou une requête SQL libre) capable de renvoyer le résultat se trouvant dans la Figure 2.

	EN TETE	TableB_5	TableB_6	TableB_7	TableB_8	TableB_9
1	TableA_1	TableC_1_5	TableC_1_6	TableC_1_7	TableC_1_8	TableC_1_9
2	TableA_2	[NON DISPONIBLE]	TableC_2_6	TableC_2_7	TableC_2_8	TableC_2_9
3	TableA_3	TableC_3_5	[NON DISPONIBLE]	TableC_3_7	TableC_3_8	TableC_3_9
4	TableA_4	TableC_4_5	TableC_4_6	TableC_4_7	TableC_4_8	TableC_4_9

Figure 2

Mode de licence de « The SQL Server 2-D Matrix Builder »

[TO BE DEFINED]

Installation de « The SQL Server 2-D Matrix Builder »

Ce projet a été développé en utilisant l'une des fonctionnalités de SQL Server 2005 les plus importantes pour les développeurs, à savoir l'intégration de la plateforme .NET directement au niveau du moteur de la base de données (SQLCLR).

Avant de démarrer l'installation, vous devez avoir autorisé l'utilisation de .NET dans votre instance SQL Server. La Figure 3 montre comment faire cela.

```
exec sp_configure 'CLR Enabled', 1
reconfigure
```

Figure 3

Pour pouvoir déployer dans votre base de données SQL Server, l'Assembly .NET qui a été développée, il faut exécuter (en tant que *system administrators* [sa]) le code T-SQL se trouvant dans la Figure 4.

```
Alter Database VotreDatabase
Set Trustworthy On
GO

Use VotreDatabase
GO

CREATE ASSEMBLY [TheSQLServerMatrix]
FROM '<votre chemin vers la DLL>\TheMatrixSqlClr.dll'
WITH PERMISSION_SET=EXTERNAL_ACCESS
GO

CREATE TYPE [QueryBuilder]
EXTERNAL NAME [TheSQLServerMatrix].[TheMatrixSqlClr.QueryBuilder]
GO

CREATE PROCEDURE [TheMatrix]
    @rowsQuery [QueryBuilder]
    , @rowsQuery_ValueMember nvarchar(max)
    , @rowsQuery_DisplayMember nvarchar(max)
    , @firstColumnHeader nvarchar(max)
    , @columnsQuery [QueryBuilder]
    , @columnsQuery_ValueMember nvarchar(max)
    , @columnsQuery_DisplayMember nvarchar(max)
    , @contentQuery [QueryBuilder]
    , @contentQuery_ColumnValueMember nvarchar(max)
    , @contentQuery_RowValueMember nvarchar(max)
    , @contentQuery_DisplayName nvarchar(max)
    , @contentQuery_MissingValue sql_variant
    , @diagnose bit = 0
AS
EXTERNAL NAME [TheSQLServerMatrix].[TheMatrixSqlClr.StoredProcedures].[TheMatrix]
```

Figure 4

Il faut noter que cette assembly a été signée par son développeur, *Pascal Belaud*. Veillez bien à le vérifier dans les propriétés de la DLL (Figure 5, Figure 6 et Figure 7).

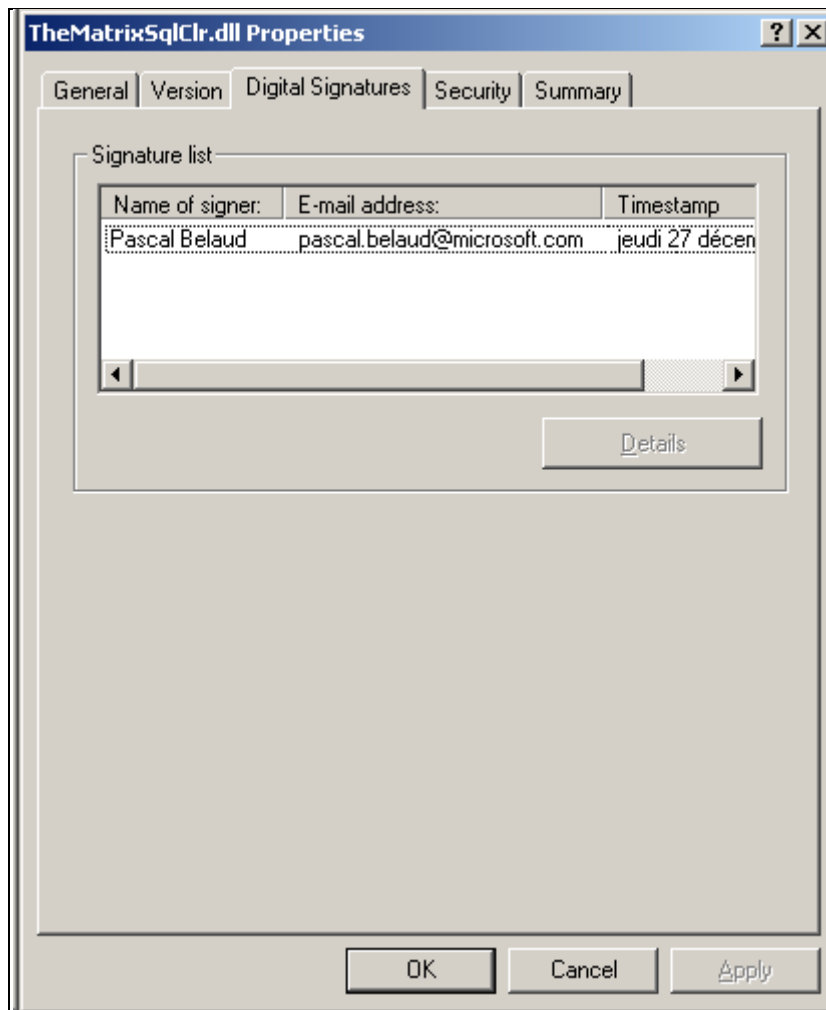


Figure 5

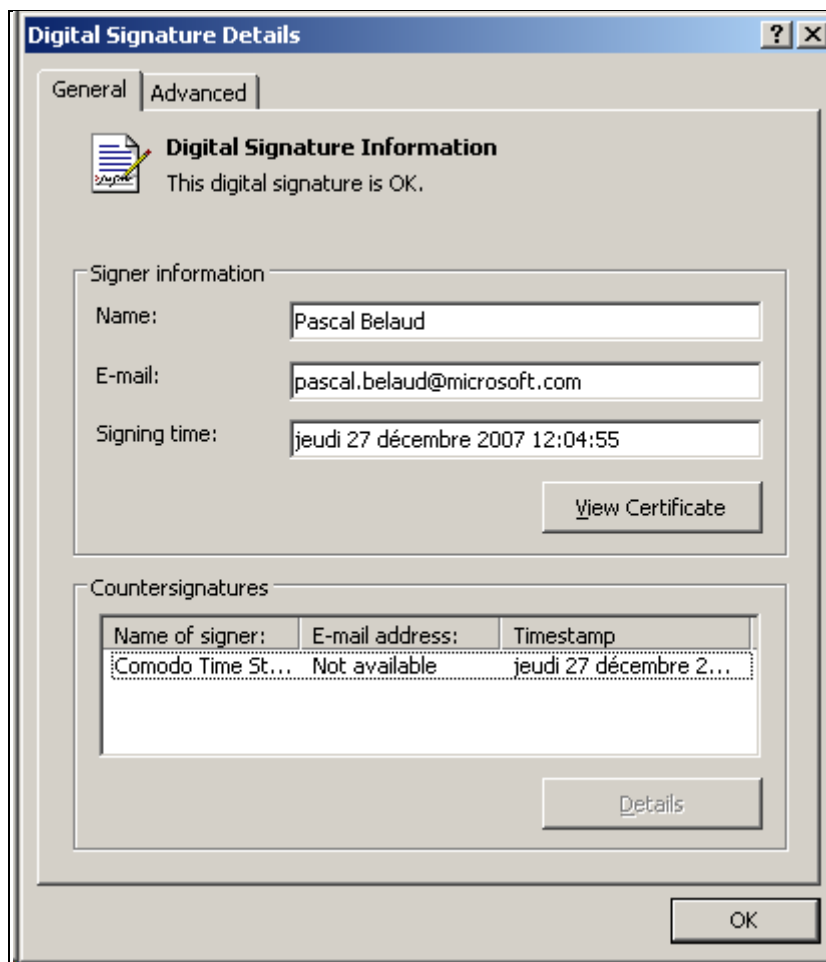


Figure 6

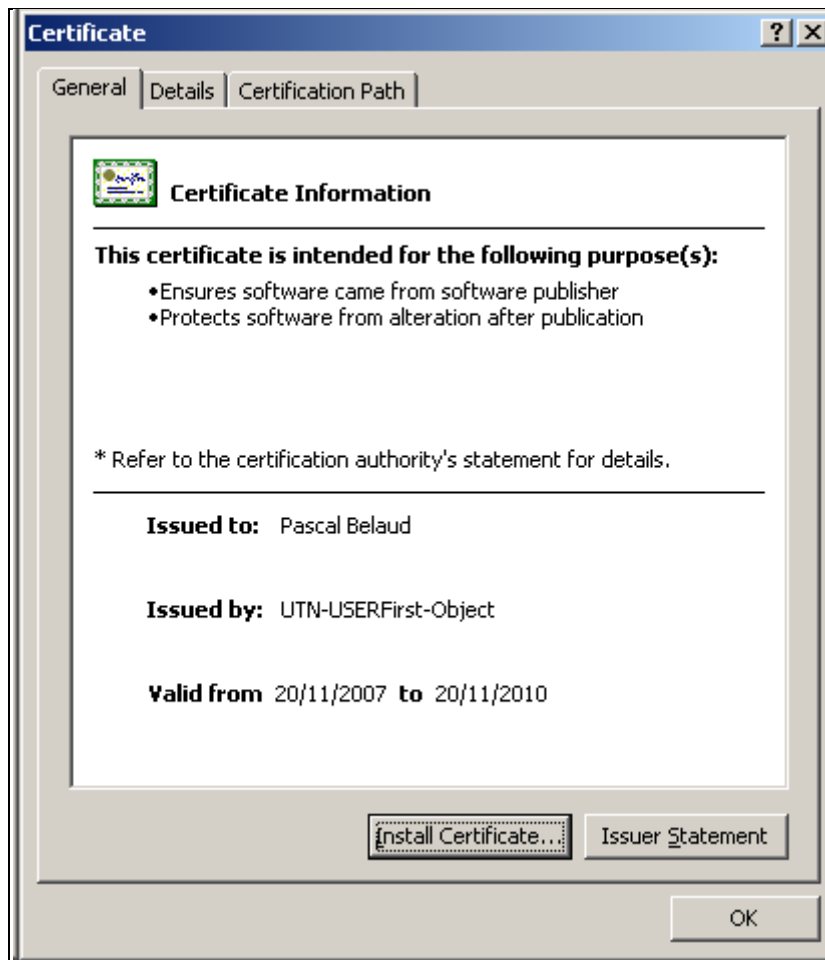


Figure 7

Vue d'ensemble de l'utilisation de « The SQL Server 2-D Matrix Builder »

Nous verrons dans les trois exemples fournis comment utiliser en détail la procédure stockée [TheMatrix]. Nous allons décrire les paramètres attendus par cette procédure stockée (Figure 8).

```

CREATE PROCEDURE [TheMatrix]
    @rowsQuery [QueryBuilder]
    , @rowsQuery_ValueMember nvarchar(max)
    , @rowsQuery_DisplayMember nvarchar(max)
    , @firstColumnHeader nvarchar(max)
    , @columnsQuery [QueryBuilder]
    , @columnsQuery_ValueMember nvarchar(max)
    , @columnsQuery_DisplayMember nvarchar(max)
    , @contentQuery [QueryBuilder]
    , @contentQuery_ColumnValueMember nvarchar(max)
    , @contentQuery_RowValueMember nvarchar(max)
    , @contentQuery_DisplayName nvarchar(max)
    , @contentQuery_MissingValue sql_variant
    , @diagnose bit = 0

```

Figure 8

Comme vous pouvez le constater, il va falloir définir les trois requêtes à utiliser pour décrire :

- Les en-têtes de colonnes à utiliser
- Les lignes se trouvant dans la première colonne
- Le contenu de la matrice

Pour ce faire, nous allons utiliser un *User Defined Type* spécifique qui a également été développé en .NET pour l'occasion : *QueryBuilder* (Figure 9)

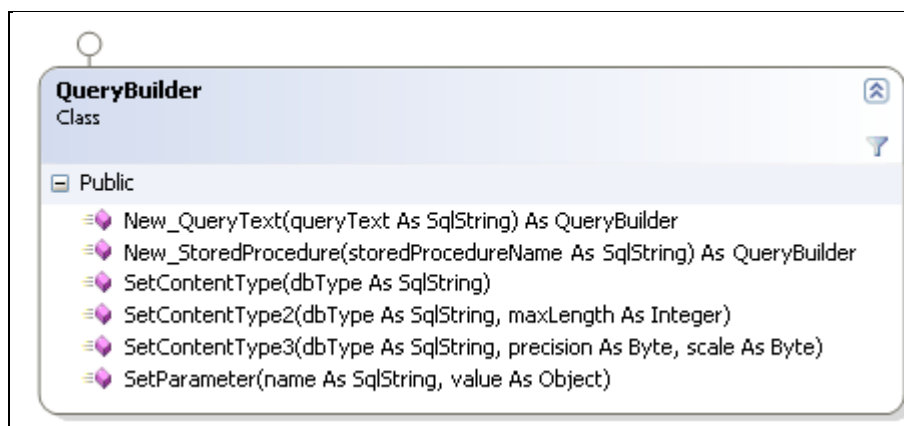


Figure 9

Ce type va nous permettre de spécifier quel type de requête nous souhaitons exécuter (texte SQL libre ou procédure stockée). Nous allons également avoir la possibilité de déclarer les éventuels paramètres d'entrée nécessaire. Enfin, nous allons également avoir la possibilité de spécifier précisément le type qui est renvoyé par la requête.

Pour indiquer que nous souhaitons exécuter du texte SQL libre, nous devons invoquer, en T-SQL, la méthode statique *New_QueryText* tandis que si nous souhaitons exécuter une procédure stockée, c'est la méthode statique *New_StoredProcedure* qui doit être invoquée (Figure 10).

```
Declare @MaRequeteTexteLibre QueryBuilder
Set @MaRequeteTexteLibre = QueryBuilder::New_QueryText(N'Votre requête SQL ici')

Declare @MaRequeteProcédureStockée QueryBuilder
Set @MaRequeteProcédureStockée = QueryBuilder::New_StoredProcedure(N'Le nom de votre SP ici')
```

Figure 10

Si vos requêtes nécessitent des paramètres en entrée, il est possible de spécifier ceux-ci en invoquant la méthode *SetParameter* (Figure 11).

```
@MaRequeteTexteLibre.SetParameter('@MonParam1', 'MaValeur1')
@MaRequeteTexteLibre.SetParameter('@MonParam2', 'MaValeur2')
@MaRequeteTexteLibre.SetParameter('@MonParam3', 'MaValeur3')
```

Figure 11

Enfin, *[TheMatrix]* va être capable d'inférer automatiquement le type à utiliser pour gérer le retour de l'exécution de vos procédures stockées. Cependant, dans certains cas, cette inférence de type peut ne pas être judicieuse dans votre cas. Vous pouvez donc imposer le type à utiliser en invoquant les méthodes *SetContentType*, *SetContentType2* et *SetContentType3*. La X montre un exemple d'utilisation de chacune des trois méthodes.

```
@MaRequeteTexteLibre.SetContentType(N'Bit')
@MaRequeteTexteLibre.SetContentType2(N'NVarChar', 255)
@MaRequeteTexteLibre.SetContentType3(N'Decimal', 9, 2)
```

Figure 12

Les valeurs adéquates pour le paramètre *dbtype* sont l'une des valeurs de l'énumération *.NET System.Data.SqlDbType* (Figure 13).

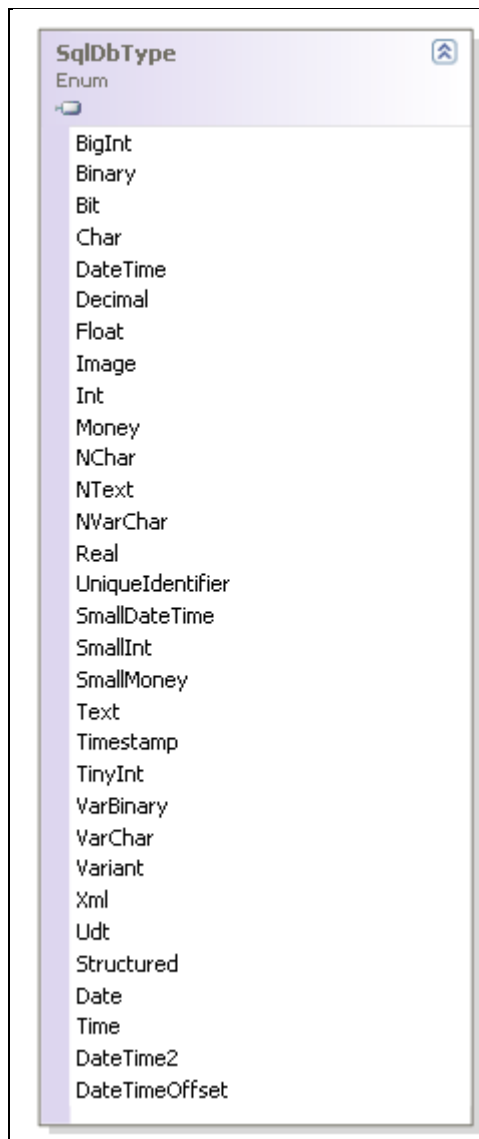


Figure 13

A noter qu'enfin il est possible d'exécuter cette requête avec le mode diagnostic activé afin de vérifier que la procédure *[TheMatrix]* a bien compris votre intention. Pour cela, il suffit de positionner le paramètre *@diagnose* à 1. Dans ce cas, un log détaillé apparaîtra dans l'onglet « Messages » (Figure 14).

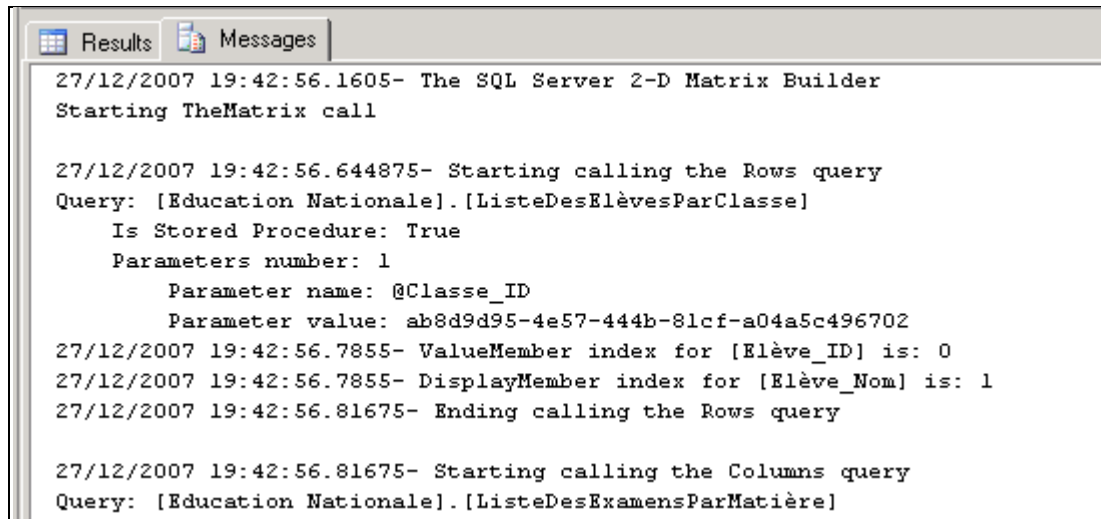


Figure 14

Exemple d'utilisation n°1 : cas simple

Imaginons que nous gèrions la vie d'un collège et notamment des notions telles que :

- Les classes.
- Les matières.
- Les coefficients des matières enseignées dans telle ou telle classe.

Il est fort probable que ceci soit ramené à la création de trois tables dans notre base de données, une pour gérer les classes, une pour gérer les matières enseignées au collège et enfin une permettant de relier une classe à une matière et qui permet de saisir le coefficient affecté à la matière pour la classe en question (Figure 15).

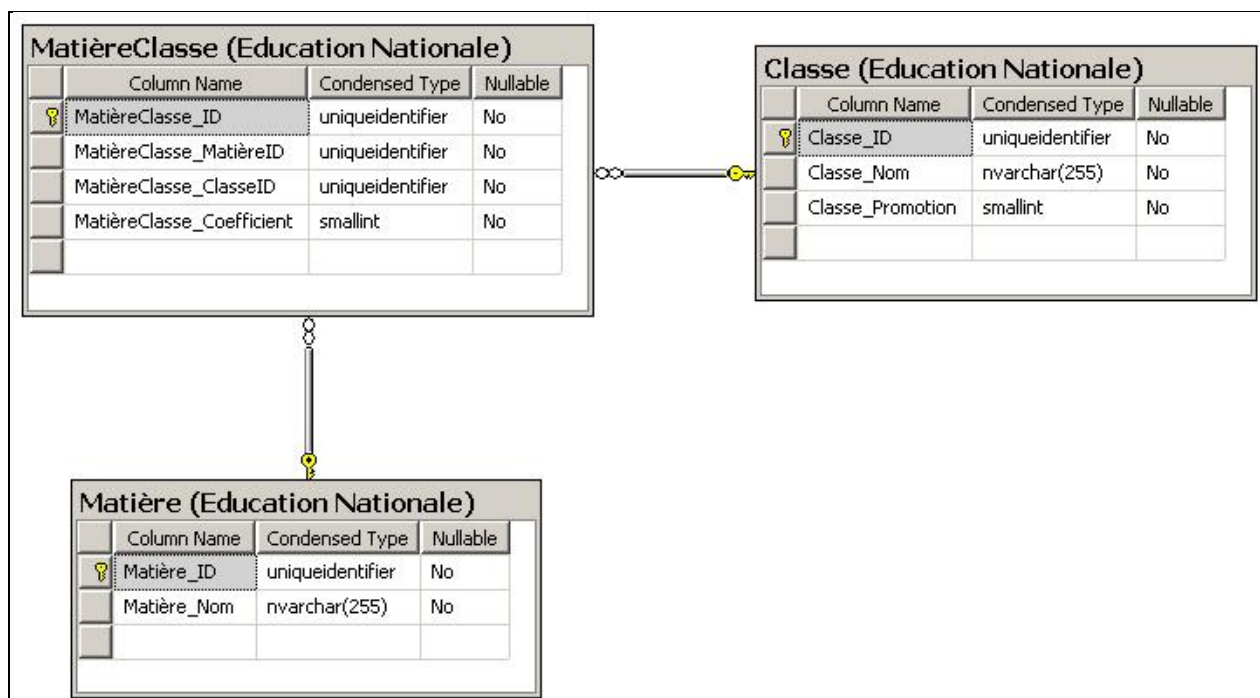


Figure 15

L'objectif de ce projet est de bénéficier d'une procédure stockée capable de nous renvoyer les données précédentes sous la forme de la Figure 16.

Results		Messages									
	Matière	6°A	6°B	6°C	5°A	5°B	5°C	4°A	4°B	3°A	3°B
1	Allemand	NULL	2	NULL	NULL	2	NULL	NULL	2	NULL	2
2	Anglais	2	NULL	NULL	2	NULL	NULL	2	NULL	2	NULL
3	Chimie	3	3	3	3	3	3	3	3	3	3
4	Espagnol	NULL	NULL	2	NULL	NULL	2	NULL	NULL	NULL	NULL
5	Français	3	3	3	3	3	3	3	3	3	3
6	Histoire/Géographie	2	2	2	2	2	2	2	2	2	2
7	Mathématiques	4	4	4	4	4	4	4	4	4	4
8	Musique	1	1	1	1	1	1	1	1	1	1
9	Physique	3	3	3	3	3	3	3	3	3	3
10	Sport	1	1	1	1	1	1	1	1	1	1

Figure 16

On voit que la matrice renvoyée ici est totalement dynamique au niveau des lignes mais également, **et surtout**, au niveau des colonnes. On constate également que certaines cellules de la matrice ne sont pas remplies car, par exemple, la matière Allemand n'est pas enseignée à la classe de 6°A et qu'en lieu et place de la valeur manquante, nous voyons qu'un champ *NULL* a été affecté.

Concernant les colonnes renvoyées, on constate qu'elles sont de deux catégories :

- La première colonne (="Matière"), qui indique ici le type de valeur que l'on trouve dans la première colonne.
- Le reste des colonnes, qui correspondent au contenu de la table *[Education Nationale].[Classe]* trié dans un ordre spécifique.

Concernant les données renvoyées dans la première colonne, elles correspondent au contenu de la table *[Education Nationale].[Matière]* trié dans un ordre spécifique.

Enfin, le contenu de la matrice proprement dite, il correspond au contenu de la table *[Education Nationale].[MatièreClasse]* et notamment la colonne *MatièreClasse_Coefficient*, qui est la valeur réellement affichée dans la matrice.

Si l'on ramène cela à des requêtes SQL, cela nous donne les requêtes visibles dans la Figure 17:

```

-- Requête pour obtenir les colonnes
Select Classe_ID, Classe_Nom

From [Education Nationale].[Classe]

Order By
    SubString(Classe_Nom, 1, 1) Desc
    , SubString(Classe_Nom, 3, 1) Asc

-- Requête pour obtenir les données de la première colonne uniquement
Select Matière_ID, Matière_Nom

From [Education Nationale].[Matière]

Order By Matière_Nom Asc

```

Figure 17

Il faut bien noter que, dans les deux cas, ces requêtes doivent renvoyer et une clé primaire et une valeur à afficher. Les clés primaires renvoyées serviront un peu plus tard à retrouver les bonnes valeurs pour les coefficients lors du remplissage de la matrice.

Pour définir ces requêtes au niveau de notre matrice, nous devons utiliser des variables de type *QueryBuilder*. Leur utilisation est on ne peut plus simple (Figure 18).

```

Declare @MesMatières QueryBuilder
Set @MesMatières = QueryBuilder::New_QueryText('
    Select Matière_ID, Matière_Nom
    From [Education Nationale].[Matière]
    Order By Matière_Nom
')

Declare @MesClasses QueryBuilder
Set @MesClasses = QueryBuilder::New_QueryText('
    Select Classe_ID, Classe_Nom
    From [Education Nationale].[Classe]
    Order By
        SubString(Classe_Nom, 1, 1) Desc
        , SubString(Classe_Nom, 3, 1) Asc
')

```

Figure 18

Nous devons, dans un premier temps, invoquer la méthode statique *New_QueryText* sur notre type *QueryBuilder* pour pouvoir lui fournir les requêtes, sous forme de texte, à utiliser.

Il faut maintenant mettre au point la requête qui va nous permettre de remplir le contenu de la matrice. Nous devons pour cela renvoyer un jeu d'enregistrements contenant 3 colonnes :

- La clé primaire de la classe
- La clé primaire de la matière
- Le coefficient à afficher

La Figure 19 montre un exemple d'une telle requête SQL :

```
-- Requête pour obtenir le contenu de la matrice
Select
  MatièreClasse_MatièreID           -- Clé primaire de la matière
, MatièreClasse_ClasseID           -- Clé primaire de la classe
, MatièreClasse_Coefficient As MatièreClasse_Coefficient -- Coefficient à afficher
From [Education Nationale].[MatièreClasse]
```

Figure 19

Il suffit maintenant de définir la variable de type *QueryBuilder* que l'on va utiliser pour définir cette requête. C'est ce que nous avons fait dans la Figure 20.

```
Declare @MesCoefficients QueryBuilder
Set @MesCoefficients = QueryBuilder::New_QueryText('
  Select
    MatièreClasse_MatièreID
    , MatièreClasse_ClasseID
    , MatièreClasse_Coefficient As MatièreClasse_Coefficient
  From [Education Nationale].[MatièreClasse]
')
```

Figure 20

A partir de là, nous avons quasiment tout ce qu'il faut pour déclencher l'appel à la procédure stockée qui va nous générer en retour la matrice attendue. Pour cela, nous devons appeler la procédure stockée *[TheMatrix]* et lui fournir les informations suivantes :

- La requête pour remplir la première colonne
- La valeur à afficher dans l'en-tête de cette colonne
- La requête pour remplir les en-têtes des autres colonnes
- La requête pour remplir la matrice
- La valeur à utiliser en cas de donnée manquante dans la matrice

C'est ce qui a été fait dans la Figure 21.

```

Exec [TheMatrix]

-- Description des lignes
@MesMatières -- la requête à exécuter
, N'Matière_ID' -- La colonne qui sert de clé primaire
, N'Matière_Nom' -- La colonne qui sert pour l'affichage

-- En-tête de la première colonne
, N'Matière'

-- Description des colonnes
@MesClasses -- la requête à exécuter
, N'classe_ID' -- La colonne qui sert de clé primaire
, N'classe_Nom' -- La colonne qui sert pour l'affichage

-- Description du contenu de la matrice
@MesCoefficients -- la requête à exécuter
, N'MatièreClasse_MatièreID' -- La colonne qui renvoie la clé primaire de la ligne
, N'MatièreClasse_classeID' -- La colonne qui renvoie la clé primaire de la colonne
, N'MatièreClasse_Coefficient' -- La colonne à afficher dans la matrice

-- Valeur de remplacement en cas d'absence de données
, NULL

```

Figure 21

L'exécution de cette requête donne lieu à la création de la matrice présentée à la Figure 22.

	Matière	6°A	6°B	6°C	5°A	5°B	5°C	4°A	4°B	3°A	3°B
1	Allemand	NULL	2	NULL	NULL	2	NULL	NULL	2	NULL	2
2	Anglais	2	NULL	NULL	2	NULL	NULL	2	NULL	2	NULL
3	Chimie	3	3	3	3	3	3	3	3	3	3
4	Espagnol	NULL	NULL	2	NULL	NULL	2	NULL	NULL	NULL	NULL
5	Français	3	3	3	3	3	3	3	3	3	3
6	Histoire/Géographie	2	2	2	2	2	2	2	2	2	2
7	Mathématiques	4	4	4	4	4	4	4	4	4	4
8	Musique	1	1	1	1	1	1	1	1	1	1
9	Physique	3	3	3	3	3	3	3	3	3	3
10	Sport	1	1	1	1	1	1	1	1	1	1

Figure 22

Exemple d'utilisation n°2 : cas complexe

Dans la base de données précédente, nous avons également des tables nous permettant de représenter la notion d'élèves, d'examen et de notes obtenues à ces examens (Figure 23).

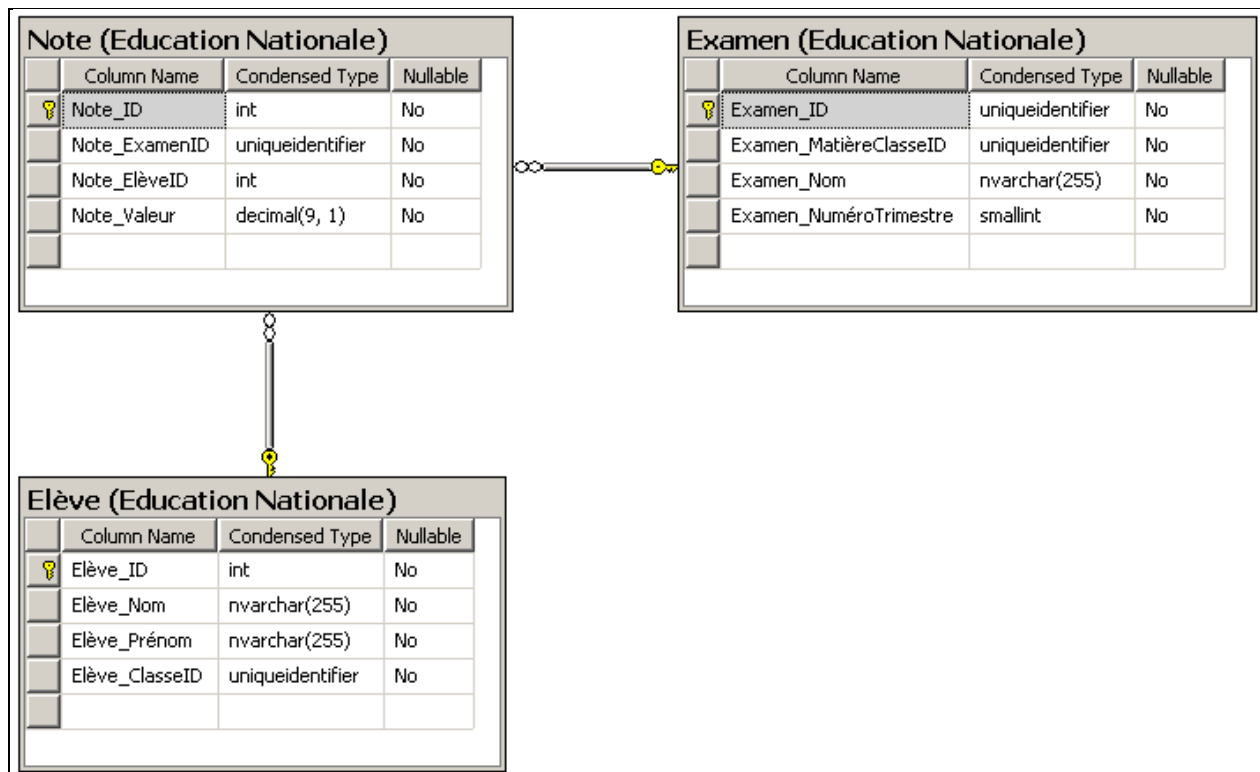


Figure 23

L'idée désormais est de construire une matrice permettant de voir l'ensemble des notes des élèves d'une classe donnée, pour une matière donnée et pour un trimestre donné. La Figure 24 montre le type de retour que nous attendons.

Results Messages				
	Mathématiques [1er trimestre 2007]	Examen 06	Examen 07	Examen 08
1	André Pierre Erol	6.00	14.50	4.50
2	Belaud Laurent	8.50	18.50	6.00
3	Belser Cyrielle	2.50	14.50	13.50
4	Beraud Damien	7.50	1.00	13.50
5	Billon Christian	8.00	0.00	6.50
6	Bonnot Samuel	14.00	14.50	19.00

Figure 24

Pour faire ceci, nous allons mélanger des appels à des procédures stockées mais également à des requêtes SQL de type texte. De plus, l'ensemble des appels que nous allons effectuer nécessite la fourniture de plusieurs paramètres en entrée, ce qui n'était pas le cas dans l'exemple n°1.

La liste des élèves de la première colonne est fournie par la procédure stockée *[ListeDesElèvesParClasse]* (Figure 25), procédure qui prend le paramètre *@Classe_ID* en entrée correspondant à l'identifiant de la classe pour laquelle nous souhaitons la liste des élèves.

```
Create Procedure [Education Nationale].[ListeDesElèvesParClasse]
(
    @Classe_ID uniqueidentifier
)
As
Select
    Elève_ID
, Elève_Nom + ' ' + Elève_Prénom As Elève_Nom
From [Education Nationale].Elève
Where Elève_ClasseID = @Classe_ID
Order By Elève_Nom Asc, Elève_Prénom Asc
Return @@RowCount
```

Figure 25

La liste des colonnes correspond à la liste des examens effectués pour une matière donnée et un trimestre donnée. Celle-ci est fournie par la procédure stockée *[ListeDesExamensParMatière]* (Figure 26), procédure qui prend les paramètres suivants en entrée :

- *@Classe_ID* : correspond à l'identifiant de la classe pour laquelle nous souhaitons la liste des examens
- *@Matière_ID* : correspond à l'identifiant de la matière pour laquelle nous souhaitons la liste des examens
- *@Trimestre* : correspond au numéro de trimestre souhaité pour les examens

```

Create Procedure [Education Nationale].[ListeDesExamensParMatière]
(
    @Classe_ID uniqueidentifier
, @Matière_ID uniqueidentifier
, @Trimestre smallint
)
As

Select
    Examen_ID
, Examen_Nom

From [Education Nationale].Examen
    Inner Join [Education Nationale].MatièreClasse
        On MatièreClasse_ID = Examen_MatièreClasseID

Where
    MatièreClasse_MatièreID = @Matière_ID
And MatièreClasse_ClasseID = @Classe_ID
And Examen_NuméroTrimestre = @Trimestre

Order By Examen_Nom Asc

Return @@RowCount

```

Figure 26

Concernant le contenu de la matrice, son contenu est fourni par l'appel d'une requête SQL (sous forme de texte (Figure 27). Cependant, cette requête a également trois paramètres en entrée :

- *@Classe_ID* : correspond à l'identifiant de la classe pour laquelle nous souhaitons la liste des notes
- *@Matière_ID* : correspond à l'identifiant de la matière pour laquelle nous souhaitons la liste des notes
- *@Trimestre* : correspond au numéro de trimestre souhaité pour les notes

```

Select Note_ExamenID, Note_ElèveID, Note_Valeur
From [Education Nationale].Note
    Inner join [Education Nationale].Examen
        On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse
        On Examen_MatièreClasseID = MatièreClasse_ID
Where
    MatièreClasse_ClasseID = @Classe_ID
And MatièreClasse_MatièreID = @Matière_ID
And Examen_NuméroTrimestre = @Trimestre

```

Figure 27

Supposons que nous souhaitions ramener les notes de tous les élèves de 6°B (promotion 2007) en mathématique sur le deuxième trimestre de l'année. Nous allons tout d'abord définir les trois variables qui vont nous permettre de stocker ces valeurs (Figure 28).

```

Declare @Classe_ID uniqueidentifier
Set @Classe_ID = 'ab8d9d95-4e57-444b-81cf-a04a5c496702'

Declare @Matière_ID uniqueidentifier
Set @Matière_ID = 'ce6a8357-d6ca-4fef-aed6-c2b5fa8c1451'

Declare @Trimestre smallint
Set @Trimestre = 2

```

Figure 28

Nous allons ensuite définir la requête qui va être utilisée pour remplir la première colonne (Figure 29). Il faut noter l'appel à la méthode *SetParameter* de notre type *QueryBuilder*, ce qui nous permet de fournir la valeur du paramètre attendue (*@Classe_ID*) par notre procédure stockée.

```
Declare @MesElèves QueryBuilder  
  
Set @MesElèves = QueryBuilder::New_StoredProcedure(  
N'[Education Nationale].[ListeDesElèvesParClasse]')  
  
Set @MesElèves.SetParameter('@Classe_ID', @Classe_ID)
```

Figure 29

Nous allons faire de même pour la requête renvoyant la liste des examens du trimestre en question (Figure 30).

```
Declare @MesExamens QueryBuilder  
  
Set @MesExamens = QueryBuilder::New_StoredProcedure(  
N'[Education Nationale].[ListeDesExamensParMatière]')  
  
Set @MesExamens.SetParameter('@Classe_ID', @Classe_ID)  
Set @MesExamens.SetParameter('@Matière_ID', @Matière_ID)  
Set @MesExamens.SetParameter('@Trimestre', @Trimestre)
```

Figure 30

Enfin, nous allons définir la requête qui va remplir la matrice (Figure 31).

```

Declare @MesMoyennes QueryBuilder

Set @MesMoyennes = QueryBuilder::New_QueryText(
N'Select Note_ExamenID, Note_ElèveID, Note_Valeur

From [Education Nationale].Note
    Inner join [Education Nationale].Examen
        On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse
        On Examen_MatièreClasseID = MatièreClasse_ID

Where MatièreClasse_ClasseID = @Classe_ID
And MatièreClasse_MatièreID = @Matière_ID
And Examen_NuméroTrimestre = @Trimestre')

Set @MesMoyennes.SetParameter('@Classe_ID', @Classe_ID)
Set @MesMoyennes.SetParameter('@Matière_ID', @Matière_ID)
Set @MesMoyennes.SetParameter('@Trimestre', @Trimestre)

```

Figure 31

A noter qu'ici, nous avons aussi la possibilité de définir nous-mêmes le type des cellules de la matrice. Si on ne précise rien, une inférence du type sera effectuée. ATTENTION, cette inférence peut ne pas utiliser le type le plus approprié à votre situation. C'est la raison pour laquelle, il peut être utile de spécifier nous-mêmes ce type (Figure 32).

```

Set @MesMoyennes.SetContentType3(N'Decimal', 9, 2)

```

Figure 32

Désormais, nous pouvons appeler la procédure stockée *[TheMatrix]* comme indiqué à la Figure 33.

```

Exec [TheMatrix]

-- Description des lignes
@MesElèves      -- la requête à exécuter
, N'Elève_ID'    -- La colonne qui sert de clé primaire
, N'Elève_Nom'   -- La colonne qui sert pour l'affichage

-- En-tête de la première colonne
, N'Mathématiques [1er trimestre 2007]'

-- Description des colonnes
, @MesExamens    -- la requête à exécuter
, N'Examen_ID'   -- La colonne qui sert de clé primaire
, N'Examen_Nom'  -- La colonne qui sert pour l'affichage

-- Description du contenu de la matrice
, @MesMoyennes   -- la requête à exécuter
, N'Note_ElèveID' -- La colonne qui renvoie la clé primaire de la ligne
, N'Note_ExamenID' -- La colonne qui renvoie la clé primaire de la colonne
, N'Note_Valeur'  -- La colonne à afficher dans la matrice

-- Valeur de remplacement en cas d'absence de données
, NULL

```

Figure 33

L'exécution de cette procédure renvoie le résultat de la Figure 34.

Results		Messages				
	Mathématiques [1er trimestre 2007]	Examen 06	Examen 07	Examen 08	Examen 09	Examen 10
1	André Pierre Erol	6.00	14.50	4.50	1.00	5.50
2	Belaud Laurent	8.50	18.50	6.00	0.00	11.50
3	Belser Cyrielle	2.50	14.50	13.50	19.50	0.00
4	Beraud Damien	7.50	1.00	13.50	7.50	5.00
5	Billon Christian	8.00	0.00	6.50	12.00	1.50
6	Bonnot Samuel	14.00	14.50	19.00	17.00	5.00
7	Carbonati Alain	19.00	0.00	0.50	12.00	15.50
8	Cardon Stéphane	3.50	4.00	14.00	2.50	3.50
9	Chesné Xavier	14.50	13.00	5.00	7.50	11.50
10	Couzy Frank	7.50	13.00	14.00	9.00	5.50
11	Decret Gwénaél	18.00	2.00	10.00	17.00	9.50
12	Descoins Christian	15.50	1.50	6.00	19.50	10.00
13	Donati Marcel	17.00	5.00	6.00	16.50	16.00
14	Fedotoff Blaise	7.50	12.50	18.50	2.00	1.00
15	Fontaine Frédéric	3.00	0.50	8.00	16.00	18.00
16	Giacopino David	5.50	12.50	4.50	18.00	14.00
17	Herbulot Benjamin	0.00	8.50	15.00	14.00	7.00
18	Hougardy Maxence	9.50	12.50	19.00	10.50	15.00
19	Jaber Matthieu	7.50	7.00	3.00	16.00	19.50
20	Lagarde Hervé	5.50	1.00	9.00	9.00	12.50
21	Levy Christian-Pierre	2.00	16.50	13.50	17.00	12.50
22	Meijer Robert	5.50	5.00	6.00	12.50	3.50
23	Nlepe Anne-Sophie	13.50	1.50	14.00	14.00	19.00
24	Sacksteder Jean-Pierre	13.00	7.50	14.50	12.50	16.50
25	Saint Martin Blaise	4.50	18.50	17.50	7.00	8.00
26	Schmidt Damien	4.50	7.50	2.50	10.00	14.00
27	Voisin Stanislas	14.50	5.00	11.50	5.50	10.00

Figure 34

Exemple d'utilisation n°3 : cas encore plus complexe

Dans cet exemple, nous allons partir des tables qui ont été utilisées dans les deux exemples précédents (Figure 35).

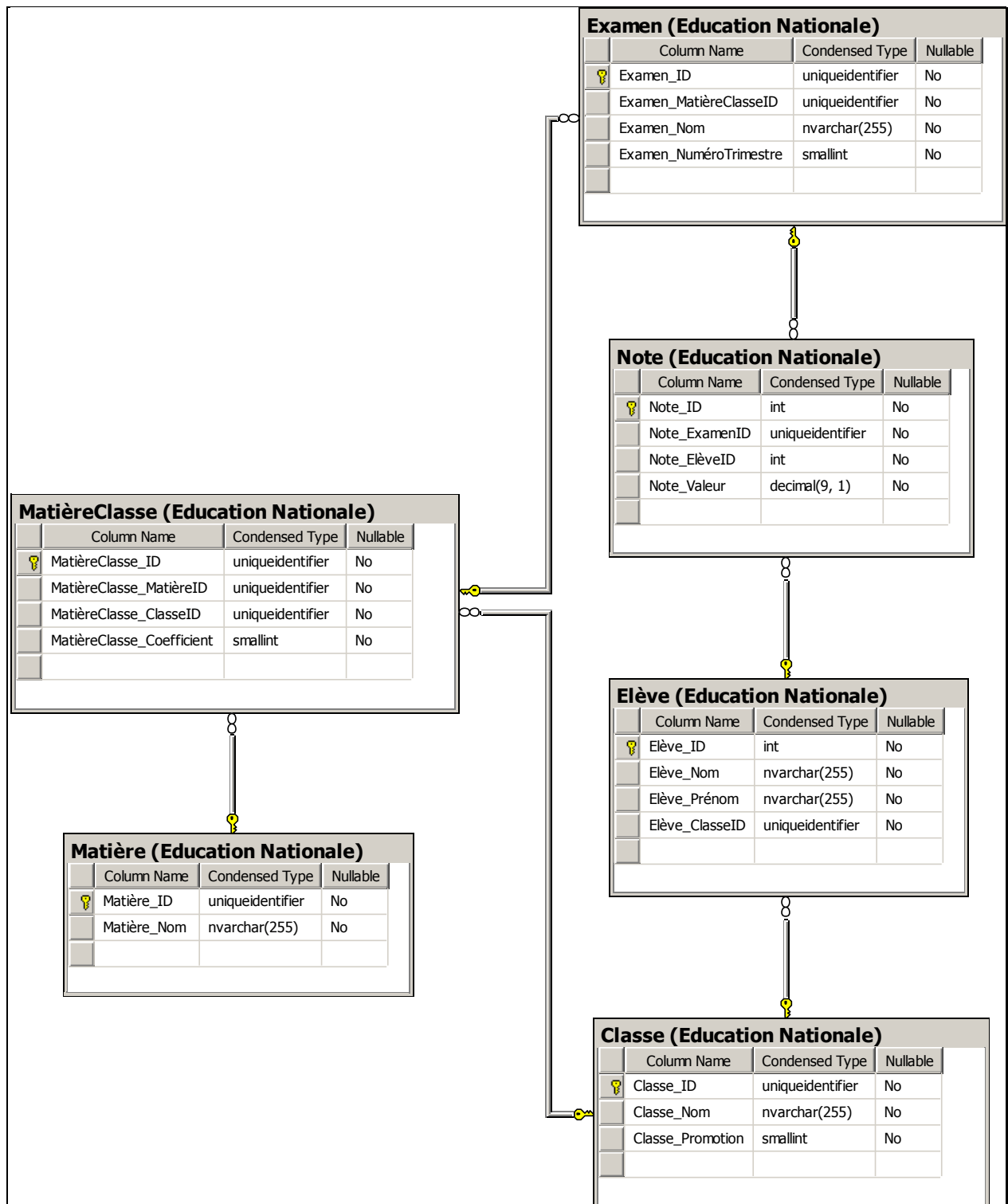


Figure 35

Nous voulons ici construire une matrice très riche. Celle-ci devra contenir, pour une classe donnée, la moyenne calculée de tous les examens pour chacun des élèves. Nous voulons de plus la moyenne du premier trimestre, du second trimestre, du troisième trimestre et de l'année pour chacun des élèves. Ce

n'est pas tout !! Nous souhaitons également voir la moyenne du premier trimestre, du second trimestre, du troisième trimestre et de l'année pour chacune des matières, tous les élèves confondus. Bien qu'en d'apparence complexe, la réalisation d'une telle matrice va être un jeu d'enfants. La Figure 36 montre le type de sortie que nous souhaitons.

	Matière1 [M, N]	Matière2 [M, N]	Matière3 [M, N]	...	← Moyenne 1er trimestre	← Moyenne 2nd trimestre	← Moyenne 3ème trimestre	← Moyenne de l'année scolaire
Classe de A - (promo B)								
Elève 1	X [Y]	X [Y]	X [Y]	...	X [Y]	X [Y]	X [Y]	X [Y]
Elève 2	X [Y]	X [Y]	X [Y]	...	X [Y]	X [Y]	X [Y]	X [Y]
Elève 3	X [Y]	X [Y]	X [Y]	...	X [Y]	X [Y]	X [Y]	X [Y]
...
Moyenne 1er trimestre ->	X	X	X	...				
Moyenne 2nd trimestre ->	X	X	X	...				
Moyenne 3ème trimestre ->	X	X	X	...				
Moyenne de l'année scolaire ->	X	X	X	...				
	A = nom complet de la classe							
	B = année de promotion							
	M = Nombre total d'exams de la matière							
	N = Coefficient de la matière dans cette classe							
	X = Moyenne obtenue							
	Y = Nombre d'exams utilisés dans le calcul de X							

Figure 36

La complexité (relative !) de cet exemple par rapport à l'exemple n°2 est que nous allons devoir mélanger des données provenant de la base de données (comme *Elève 1*, *Elève 2*...), avec des données non présentes dans la base comme (*Moyenne 1er trimestre ->*, *Moyenne 2nd trimestre ->*, ...). Il va donc falloir « affecter » des clés primaires à ces pseudos enregistrements et les faire passer comme des élèves pour le contenu de la première colonne et comme des matières pour la liste des en-têtes de colonne de la matrice. Nous allons donc fixer des clés primaires pour ces 8 enregistrements supplémentaires (les 4 dernières lignes et les 4 derniers en-têtes de colonne).

Le type de clé primaire des élèves est *[int]*, tandis que celui des matières est *[uniqueidentifiser]* (Figure 37).

"Elèves"	ID					
Moyenne 1er trimestre ->	-1					
Moyenne 2nd trimestre ->	-2					
Moyenne 3ème trimestre ->	-3					
Moyenne de l'année scolaire ->	-4					
"Matières"						
<- Moyenne 1er trimestre	{0FF9912B-67DC-44de-B886-9A56CDAEAE3}					
<- Moyenne 2nd trimestre	{6B14C9A3-7114-48f0-9D71-B205201A801E}					
<- Moyenne 3ème trimestre	{DECB8711-A6BA-4a1a-9277-58E827D7911F}					
<- Moyenne de l'année scolaire	{F36E9B66-6B07-4ba5-88F3-957638DFDC96}					

Figure 37

Il ne nous reste plus maintenant qu'à définir les trois procédures stockées qui vont être appelées pour construire la matrice attendue :

- [ListeDesElèvesParClasseAvecMoyenne] (Figure 38) pour la liste des élèves
- [ListeDesMatièresParClasse] (Figure 39) pour la liste des matières
- [CalculDesMoyennesParClasse] (de Figure 40 à Figure 46) pour le calcul des moyennes

Le contenu même de ces trois procédures n'est pas vraiment intéressant ici. Il faut juste remarquer l'utilisation de table temporaire pour pouvoir stocker à la fois les valeurs en provenance de la base de données mais également les valeurs fictives décrites précédemment.

```

Create Procedure [Education Nationale].[ListeDesElèvesParClasseAvecMoyenne]
(
    @Classe_ID uniqueidentifier
)
As
Declare @Elève Table (
    ID int
    , Nom nvarchar(255)
    , Ordre int identity(1,1)
)
Insert Into @Elève
    Select Elève_ID
        , Elève_Nom + ' ' + Elève_Prénom
    From [Education Nationale].Elève
    Where Elève_ClasseID = @Classe_ID
    Order By Elève_Nom Asc, Elève_Prénom Asc
Insert Into @Elève VALUES (-1, N'Moyenne 1er trimestre ->')
Insert Into @Elève VALUES (-2, N'Moyenne 2nd trimestre ->')
Insert Into @Elève VALUES (-3, N'Moyenne 3ème trimestre ->')
Insert Into @Elève VALUES (-4, N'Moyenne de l'année scolaire ->')
Select
    ID As Elève_ID
    , Nom As Elève_Nom
From @Elève
Order By Ordre
Return @@RowCount

```

Figure 38

```

Declare @Matiere Table (ID uniqueidentifier, Nom nvarchar(255), Ordre int identity(1,1))

Insert Into @Matiere
Select Matière_ID
, Matière_Nom + N' [' + Convert(nvarchar(2), (Select Count(*) From [Education Nationale].Examen
where Examen_MatièreClasseID = MatièreClasse_ID))
+ N', ' + Convert(nvarchar(2), MatièreClasse_Coefficient) + N']'
From [Education Nationale].MatièreClasse
Inner Join [Education Nationale].Matière
ON Matière_ID = MatièreClasse_MatièreID
Where MatièreClasse_ClasseID = @Classe_ID
Order By MatièreClasse_Coefficient Desc, Matière_Nom Asc

Insert Into @Matiere
VALUES ('{0FF9912B-67DC-44de-B886-9A56CDAAEE3}', N'<- Moyenne 1er trimestre')

Insert Into @Matiere
VALUES ('{6B14C9A3-7114-48f0-9D71-B205201A801E}', N'<- Moyenne 2nd trimestre')

Insert Into @Matiere
VALUES ('{DECB8711-A6BA-4a1a-9277-58E827D7911F}', N'<- Moyenne 3ème trimestre')

Insert Into @Matiere
VALUES ('{F36E9B66-6B07-4ba5-88F3-957638DFDC96}', N'<- Moyenne de l'année scolaire')

Select
ID As Matière_ID
, Nom As Matière_Nom

From @Matiere Order By Ordre

Return @@RowCount

```

Figure 39

```

Create Procedure [Education Nationale].[CalculDesMoyennesParClasse]
(
@Classe_ID uniqueidentifier
)
As

Declare @MoyennePremierTrimestreParElève_ID uniqueidentifier
Declare @MoyenneSecondTrimestreParElève_ID uniqueidentifier
Declare @MoyenneTroisièmeTrimestreParElève_ID uniqueidentifier
Declare @MoyenneGénéraleParElève_ID uniqueidentifier

Set @MoyennePremierTrimestreParElève_ID = '{0FF9912B-67DC-44de-B886-9A56CDAAEE3}'
Set @MoyenneSecondTrimestreParElève_ID = '{6B14C9A3-7114-48f0-9D71-B205201A801E}'
Set @MoyenneTroisièmeTrimestreParElève_ID = '{DECB8711-A6BA-4a1a-9277-58E827D7911F}'
Set @MoyenneGénéraleParElève_ID = '{F36E9B66-6B07-4ba5-88F3-957638DFDC96}'

Declare @MoyennePremierTrimestreParMatière_ID int
Declare @MoyenneSecondTrimestreParMatière_ID int
Declare @MoyenneTroisièmeTrimestreParMatière_ID int
Declare @MoyenneGénéraleParMatière_ID int

Set @MoyennePremierTrimestreParMatière_ID = -1
Set @MoyenneSecondTrimestreParMatière_ID = -2
Set @MoyenneTroisièmeTrimestreParMatière_ID = -3
Set @MoyenneGénéraleParMatière_ID = -4

```

Figure 40

```

Declare @SommeCoefficientPremierTrimestre int
Select @SommeCoefficientPremierTrimestre = Sum(MatièreClasse_Coefficient)
From [Education Nationale].MatièreClasse
    Inner Join [Education Nationale].Examen On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 1

Declare @SommeCoefficientSecondTrimestre int
Select @SommeCoefficientSecondTrimestre = Sum(MatièreClasse_Coefficient)
From [Education Nationale].MatièreClasse
    Inner Join [Education Nationale].Examen On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 2

Declare @SommeCoefficientTroisièmeTrimestre int
Select @SommeCoefficientTroisièmeTrimestre = Sum(MatièreClasse_Coefficient)
From [Education Nationale].MatièreClasse
    Inner Join [Education Nationale].Examen On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 3

Declare @SommeCoefficient int
Select @SommeCoefficient = Sum(MatièreClasse_Coefficient)
From [Education Nationale].MatièreClasse
    Inner Join [Education Nationale].Examen On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID

```

Figure 41

```

Declare @Moyenne Table (
    Matière_ID uniqueidentifier
    , Elève_ID int
    , Moyenne nvarchar(255)
)

Insert Into @Moyenne
Select
    MatièreClasse_MatièreID
    , Note_ElèveID
    , Convert(nvarchar(255), Convert(decimal(9,2), Avg(Note_Valeur))) + N' [' + Convert(nvarchar(255), (
        Select Count(*) From [Education Nationale].Note Note2
        Inner Join [Education Nationale].Examen Examen2
            On Note2.Note_ExamenID = Examen2.Examen_ID
        Inner Join [Education Nationale].MatièreClasse MatièreClasse2
            On Examen2.Examen_MatièreClasseID = MatièreClasse2.MatièreClasse_ID
        Where Note2.Note_ElèveID = Note.Note_ElèveID
        And MatièreClasse2.MatièreClasse_MatièreID = MatièreClasse.MatièreClasse_MatièreID
    )) + N']'
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID
Group By Note_ElèveID, MatièreClasse_MatièreID

```

Figure 42

```

Insert Into @Moyenne
Select
    @MoyennePremierTrimestreParElève_ID, Note_ElèveID
    , Convert(nvarchar(255)
        , Convert(decimal(9,2)
            , Sum(Note_Valeur * MatièreClasse_Coefficient) / @SommeCoefficientPremierTrimestre))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 1
Group By Note_ElèveID

Insert Into @Moyenne
Select @MoyenneSecondTrimestreParElève_ID, Note_ElèveID
    , Convert(nvarchar(255)
        , Convert(decimal(9,2)
            , Sum(Note_Valeur * MatièreClasse_Coefficient) / @SommeCoefficientSecondTrimestre))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 2
Group By Note_ElèveID

Insert Into @Moyenne
Select @MoyenneTroisièmeTrimestreParElève_ID, Note_ElèveID
    , Convert(nvarchar(255)
        , Convert(decimal(9,2)
            , Sum(Note_Valeur * MatièreClasse_Coefficient) / @SommeCoefficientTroisièmeTrimestre))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 3
Group By Note_ElèveID

```

Figure 43

```

Insert Into @Moyenne
Select @MoyenneGénéraleParElève_ID, Note_ElèveID
    , Convert(nvarchar(255)
        , Convert(decimal(9,2)
            , Sum(Note_Valeur * MatièreClasse_Coefficient) / @SommeCoefficient))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID
Group By Note_ElèveID

```

Figure 44

```

Insert Into @Moyenne
Select MatièreClasse_MatièreID, @MoyennePremierTrimestreParMatière_ID
, Convert(nvarchar(255), Convert(decimal(9,2), Avg(Note_Valeur)))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 1
Group By MatièreClasse_MatièreID

Insert Into @Moyenne
Select MatièreClasse_MatièreID, @MoyenneSecondTrimestreParMatière_ID
, Convert(nvarchar(255), Convert(decimal(9,2), Avg(Note_Valeur)))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 2
Group By MatièreClasse_MatièreID

Insert Into @Moyenne
Select MatièreClasse_MatièreID, @MoyenneTroisièmeTrimestreParMatière_ID
, Convert(nvarchar(255), Convert(decimal(9,2), Avg(Note_Valeur)))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID And Examen_NuméroTrimestre = 3
Group By MatièreClasse_MatièreID

Insert Into @Moyenne
Select MatièreClasse_MatièreID, @MoyenneGénéraleParMatière_ID
, Convert(nvarchar(255), Convert(decimal(9,2), Avg(Note_Valeur)))
From [Education Nationale].Note
    Inner Join [Education Nationale].Examen On Note_ExamenID = Examen_ID
    Inner Join [Education Nationale].MatièreClasse On Examen_MatièreClasseID = MatièreClasse_ID
Where MatièreClasse_ClasseID = @Classe_ID
Group By MatièreClasse_MatièreID

```

Figure 45

```

Select
    Matière_ID
,   Elève_ID
,   Moyenne

From @Moyenne

Return @@RowCount

```

Figure 46

Il ne reste plus qu'à écrire le code d'appel à la procédure *[TheMatrix]*. Tout d'abord, nous devons déclarer toutes les variables nécessaires à l'appel (Figure 47).


```

Declare @Classe_ID uniqueidentifier
Set @Classe_ID = 'ab8d9d95-4e57-444b-81cf-a04a5c496702'

Declare @MesElèves QueryBuilder
Set @MesElèves = QueryBuilder::New_StoredProcedure(
N' [Education Nationale].[ListeDesElèvesParClasseAvecMoyenne]')
Set @MesElèves.SetParameter('@Classe_ID', @Classe_ID)
Set @MesElèves.SetContentType2(N'NVarChar', 255)

Declare @MesMatières QueryBuilder
Set @MesMatières = QueryBuilder::New_StoredProcedure(
N' [Education Nationale].[ListeDesMatièresParClasse]')
Set @MesMatières.SetParameter(N'@Classe_ID', @Classe_ID)

Declare @MesMoyennes QueryBuilder
Set @MesMoyennes = QueryBuilder::New_StoredProcedure(
N' [Education Nationale].[CalculDesMoyennesParClasse]')
Set @MesMoyennes.SetParameter(N'@Classe_ID', @Classe_ID)

Declare @EnTete nvarchar(255)
Select @EnTete = N'classe de ' + Classe_Nom + N' - promo ' + Convert(nvarchar(4), Classe_Promotion)
From [Education Nationale].[Classe] where Classe_ID = @Classe_ID

```

Figure 47

Enfin, il suffit de réaliser l'appel (Figure 48).

```

Exec [TheMatrix]

-- Description des lignes
@MesElèves      -- la requête à exécuter
, N'Elève_ID'    -- La colonne qui sert de clé primaire
, N'Elève_Nom'   -- La colonne qui sert pour l'affichage

-- En-tête de la première colonne
, @EnTete

-- Description des colonnes
, @MesMatières   -- la requête à exécuter
, N'Matière_ID'  -- La colonne qui sert de clé primaire
, N'Matière_Nom' -- La colonne qui sert pour l'affichage

-- Description du contenu de la matrice
, @MesMoyennes   -- la requête à exécuter
, N'Elève_ID'    -- La colonne qui renvoie la clé primaire de la ligne
, N'Matière_ID'  -- La colonne qui renvoie la clé primaire de la colonne
, N'Moyenne'     -- La colonne à afficher dans la matrice

-- Valeur de remplacement en cas d'absence de données
, N''

```

Figure 48

La Figure 49 montre le résultat de l'exécution finale de cette procédure.

Résultats		Messages											
	Classe de 6 ^B - promo 2007	Mathémati...	Chimie [15, 3]	Français [1...	Physique [1...	Allemand [1...	Histoire/Gé...	Musique [1...	Sport [1...	<- Moyenne 1er ...	<- Moyenne 2nd...	<- Moyenne 3ème...	<- Moyenne de l'ann...
1	André Pierre Erol	9.57 [15]	8.77 [15]	8.80 [15]	12.70 [15]	8.20 [15]	5.67 [15]	11.00 [15]	9.63 [15]	9.50	9.17	9.35	9.34
2	Belaud Laurent	8.97 [15]	9.50 [15]	9.67 [15]	13.57 [15]	6.90 [15]	8.80 [15]	12.60 [15]	9.10 [15]	10.84	8.24	10.48	9.85
3	Belser Cytelle	9.73 [15]	8.57 [15]	8.77 [15]	10.93 [15]	10.30 [15]	10.70 [15]	9.90 [15]	9.50 [15]	9.04	10.24	9.95	9.74
4	Beraud Damien	7.63 [15]	7.13 [15]	10.10 [15]	8.23 [15]	11.80 [15]	11.63 [15]	8.53 [15]	11.20 [15]	8.69	9.05	9.66	9.13
5	Billon Christian	6.43 [15]	7.87 [15]	9.67 [15]	10.73 [15]	10.33 [15]	9.87 [15]	12.73 [15]	7.40 [15]	8.43	9.46	9.12	9.00
6	Bonnot Samuel	10.20 [15]	8.67 [15]	9.90 [15]	6.23 [15]	8.93 [15]	9.90 [15]	9.57 [15]	9.90 [15]	10.04	9.23	7.94	9.07
7	Carbonati Alain	9.40 [15]	8.37 [15]	9.67 [15]	8.90 [15]	10.70 [15]	7.73 [15]	8.80 [15]	10.90 [15]	9.34	9.01	9.27	9.21
8	Cardon Stéphane	9.50 [15]	9.10 [15]	11.77 [15]	9.90 [15]	9.43 [15]	11.87 [15]	8.40 [15]	8.43 [15]	10.38	9.21	10.37	9.99
9	Chesné Xavier	8.50 [15]	11.97 [15]	10.33 [15]	12.40 [15]	8.23 [15]	11.13 [15]	8.87 [15]	12.93 [15]	10.73	11.04	9.59	10.45
10	Couzy Frank	10.47 [15]	9.97 [15]	9.70 [15]	10.77 [15]	11.97 [15]	8.57 [15]	13.40 [15]	4.43 [15]	8.49	11.02	10.82	10.11
11	Decret Guénélal	7.33 [15]	12.87 [15]	12.07 [15]	11.67 [15]	11.37 [15]	10.37 [15]	9.80 [15]	9.77 [15]	10.40	11.83	9.69	10.64
12	Descours Christian	10.47 [15]	8.73 [15]	12.30 [15]	5.40 [15]	8.70 [15]	11.33 [15]	9.73 [15]	13.00 [15]	9.29	9.16	10.59	9.68
13	Donati Marcel	10.57 [15]	7.83 [15]	9.43 [15]	9.67 [15]	10.63 [15]	7.97 [15]	11.93 [15]	8.30 [15]	8.88	10.42	9.19	9.50
14	Fedotoff Blaise	11.40 [15]	11.33 [15]	10.13 [15]	14.37 [15]	9.13 [15]	10.67 [15]	12.43 [15]	6.43 [15]	10.50	11.56	11.34	11.14
15	Fontaine Frédéric	8.33 [15]	10.50 [15]	8.47 [15]	8.50 [15]	10.77 [15]	9.93 [15]	12.53 [15]	8.93 [15]	8.27	10.21	9.73	9.40
16	Giacomino David	9.17 [15]	11.37 [15]	8.93 [15]	10.90 [15]	8.30 [15]	9.27 [15]	9.20 [15]	9.53 [15]	8.87	10.13	10.08	9.69
17	Herbulot Benjamin	9.20 [15]	7.43 [15]	9.00 [15]	11.03 [15]	7.30 [15]	9.13 [15]	10.90 [15]	11.10 [15]	10.29	8.47	8.72	9.16
18	Hougardy Maxence	10.70 [15]	6.63 [15]	9.53 [15]	10.50 [15]	11.13 [15]	9.27 [15]	10.47 [15]	8.37 [15]	9.29	10.32	9.19	9.60
19	Jaber Matthieu	8.63 [15]	8.70 [15]	7.13 [15]	7.77 [15]	7.83 [15]	9.97 [15]	10.57 [15]	10.57 [15]	8.71	9.15	8.64	8.53
20	Lagarde Hervé	7.70 [15]	10.87 [15]	9.40 [15]	9.77 [15]	7.50 [15]	9.07 [15]	7.87 [15]	8.67 [15]	9.68	8.33	8.93	8.98
21	Levy Christian-Pierre	10.87 [15]	10.03 [15]	10.30 [15]	8.30 [15]	11.77 [15]	11.33 [15]	10.13 [15]	9.00 [15]	11.22	9.57	9.95	10.25
22	Meijer Robert	8.67 [15]	9.87 [15]	10.00 [15]	10.20 [15]	6.63 [15]	12.33 [15]	8.23 [15]	11.27 [15]	10.03	8.58	10.17	9.59
23	Niepe Anne-Sophie	12.27 [15]	10.13 [15]	8.53 [15]	9.80 [15]	8.80 [15]	8.20 [15]	10.73 [15]	12.10 [15]	10.53	10.13	9.55	10.07
24	Sacksteder Jean-Pierre	10.43 [15]	10.77 [15]	9.50 [15]	10.03 [15]	9.27 [15]	11.40 [15]	10.60 [15]	11.13 [15]	10.52	9.79	10.59	10.30
25	Saint Martin Blaise	8.90 [15]	8.03 [15]	9.57 [15]	11.13 [15]	10.47 [15]	9.27 [15]	7.83 [15]	11.77 [15]	8.94	9.51	10.49	9.65
26	Schmidt Damien	7.80 [15]	10.50 [15]	9.77 [15]	7.50 [15]	8.97 [15]	10.47 [15]	11.97 [15]	10.00 [15]	8.68	9.40	9.23	9.10
27	Voisin Stanislas	9.63 [15]	11.33 [15]	8.40 [15]	10.27 [15]	12.70 [15]	11.83 [15]	10.83 [15]	11.67 [15]	9.36	10.52	11.71	10.53
28	Moyenne 1er trimestre ->	9.17	8.79	9.57	9.98	9.19	10.26	11.21	9.79				
29	Moyenne 2nd trimestre ->	9.67	9.70	9.45	10.30	9.63	9.77	9.46	9.56				
30	Moyenne 3ème trimestre ->	9.21	10.05	9.96	9.85	9.86	9.72	10.39	10.10				
31	Moyenne de l'année scolaire ->	9.35	9.51	9.66	10.04	9.56	9.91	10.35	9.82				

Figure 49

Ce qui, dans Microsoft Excel et après un petit formatage cosmétique, donne la Figure 50.

Classe de 6 ^B - promo 2007	Mathématiques [15, 4]	Chimie [15, 3]	Français [15, 3]	Physique [15, 3]	Allemand [15, 2]	Histoire/Géographie [15, 2]	Musique [15, 1]	Sport [15, 1]		<- Moyenne 1er trimestre	<- Moyenne 2nd trimestre	<- Moyenne 3ème trimestre	<- Moyenne de l'année scolaire
André Pierre Erol	9.57 [15]	8.77 [15]	8.80 [15]	12.70 [15]	8.20 [15]	5.67 [15]	11.00 [15]	9.63 [15]		9.5	9.17	9.35	9.34
Belaud Laurent	8.97 [15]	9.50 [15]	9.67 [15]	13.57 [15]	6.90 [15]	8.80 [15]	12.60 [15]	9.10 [15]		10.84	8.24	10.48	9.85
Belser Cyrielle	9.73 [15]	8.57 [15]	8.77 [15]	10.93 [15]	10.30 [15]	10.70 [15]	9.90 [15]	9.50 [15]		9.04	10.24	9.95	9.74
Beraud Damien	7.63 [15]	7.13 [15]	10.10 [15]	8.23 [15]	11.80 [15]	11.63 [15]	8.53 [15]	11.20 [15]		8.69	9.05	9.66	9.13
Billon Christian	6.43 [15]	7.87 [15]	9.67 [15]	10.73 [15]	10.33 [15]	9.87 [15]	12.73 [15]	7.40 [15]		8.43	9.46	9.12	9
Bonnot Samuel	10.20 [15]	8.67 [15]	9.90 [15]	6.23 [15]	8.93 [15]	9.90 [15]	9.57 [15]	9.90 [15]		10.04	9.23	7.94	9.07
Carbonati Alain	9.40 [15]	8.37 [15]	9.67 [15]	8.90 [15]	10.70 [15]	7.73 [15]	8.80 [15]	10.90 [15]		9.34	9.01	9.27	9.21
Cardon Stéphane	9.50 [15]	9.10 [15]	11.77 [15]	9.90 [15]	9.43 [15]	11.87 [15]	8.40 [15]	8.43 [15]		10.38	9.21	10.37	9.99
Chesné Xavier	8.50 [15]	11.97 [15]	10.33 [15]	12.40 [15]	8.23 [15]	11.13 [15]	8.87 [15]	12.93 [15]		10.73	11.04	9.59	10.45
Couzy Frank	10.47 [15]	9.97 [15]	9.70 [15]	10.77 [15]	11.97 [15]	8.57 [15]	13.40 [15]	4.43 [15]		8.49	11.02	10.82	10.11
Decret Guénaël	7.33 [15]	12.87 [15]	12.07 [15]	11.67 [15]	11.37 [15]	10.37 [15]	9.80 [15]	9.77 [15]		10.4	11.83	9.69	10.64
Descoins Christian	10.47 [15]	8.73 [15]	12.30 [15]	5.40 [15]	8.70 [15]	11.33 [15]	9.73 [15]	13.00 [15]		9.29	9.16	10.59	9.68
Donati Marcel	10.57 [15]	7.83 [15]	9.43 [15]	9.67 [15]	10.63 [15]	7.97 [15]	11.93 [15]	8.30 [15]		8.88	10.42	9.19	9.5
Fedotoff Blaise	11.40 [15]	11.33 [15]	10.13 [15]	14.37 [15]	9.13 [15]	10.67 [15]	12.43 [15]	6.43 [15]		10.5	11.56	11.34	11.14
Fontaine Frédéric	8.33 [15]	10.50 [15]	8.47 [15]	8.50 [15]	10.77 [15]	9.93 [15]	12.53 [15]	8.93 [15]		8.27	10.21	9.73	9.4
Giacopino David	9.17 [15]	11.37 [15]	8.93 [15]	10.90 [15]	8.30 [15]	9.27 [15]	9.20 [15]	9.53 [15]		8.87	10.13	10.08	9.69
Herbulot Benjamin	9.20 [15]	7.43 [15]	9.00 [15]	11.03 [15]	7.30 [15]	9.13 [15]	10.90 [15]	11.10 [15]		10.29	8.47	8.72	9.16
Hougardy Maxence	10.70 [15]	6.63 [15]	9.53 [15]	10.50 [15]	11.13 [15]	9.27 [15]	10.47 [15]	8.37 [15]		9.29	10.32	9.19	9.6
Jaber Matthieu	8.63 [15]	8.70 [15]	7.13 [15]	7.77 [15]	7.83 [15]	9.97 [15]	10.57 [15]	10.57 [15]		7.81	9.15	8.64	8.53
Lagarde Hervé	7.70 [15]	10.87 [15]	9.40 [15]	9.77 [15]	7.50 [15]	9.07 [15]	7.87 [15]	8.67 [15]		9.68	8.33	8.93	8.98
Levy Christian-Pierre	10.87 [15]	10.03 [15]	10.30 [15]	8.30 [15]	11.77 [15]	11.33 [15]	10.13 [15]	9.00 [15]		11.22	9.57	9.95	10.25
Meijer Robert	8.67 [15]	9.87 [15]	10.00 [15]	10.20 [15]	6.63 [15]	12.33 [15]	8.23 [15]	11.27 [15]		10.03	8.58	10.17	9.59
Nlepe Anne-Sophie	12.27 [15]	10.13 [15]	8.53 [15]	9.80 [15]	8.80 [15]	8.20 [15]	10.73 [15]	12.10 [15]		10.53	10.13	9.55	10.07
Sacksteder Jean-Pierre	10.43 [15]	10.77 [15]	9.50 [15]	10.03 [15]	9.27 [15]	11.40 [15]	10.60 [15]	11.13 [15]		10.52	9.79	10.59	10.3
Saint Martin Blaise	8.90 [15]	8.03 [15]	9.57 [15]	11.13 [15]	10.47 [15]	10.47 [15]	7.83 [15]	11.77 [15]		8.94	9.51	10.49	9.65
Schmidt Damien	7.80 [15]	10.50 [15]	9.77 [15]	7.50 [15]	8.97 [15]	9.27 [15]	11.97 [15]	10.00 [15]		8.68	9.4	9.23	9.1
Voisin Stanislas	9.63 [15]	11.33 [15]	8.40 [15]	10.27 [15]	12.70 [15]	11.83 [15]	10.83 [15]	11.67 [15]		9.36	10.52	11.71	10.53
Moyenne 1er trimestre ->	9.17	8.79	9.57	9.98	9.19	10.26	11.21	9.79					
Moyenne 2nd trimestre ->	9.67	9.7	9.45	10.3	9.63	9.77	9.46	9.56					
Moyenne 3ème trimestre ->	9.21	10.05	9.96	9.85	9.86	9.72	10.39	10.1					
Moyenne de l'année scolaire ->	9.35	9.51	9.66	10.04	9.56	9.91	10.35	9.82					

Figure 50

Ensuite ?

Cet article est seulement la première étape de ce projet. L'étape suivante va être la publication du code source VB .NET de ce projet sur le site de partage des projets Open Source baptisé CodePlex :

<http://www.codeplex.com>

Il y'aura également un article technique décrivant cette fois-ci l'envers du décor, à savoir comment j'ai développé ce projet.

Enfin, pour suivre les nouveautés sur ce projet, le mieux est de vous rendre sur mon blog à l'adresse suivante :

<http://blogs.msdn.com/pascal/archive/tags/The+SQL+Server+2-D+Matrix+Builder/default.aspx>