

BILAN INDIVIDUEL DE FORMATION

Période du : au :

Filière : FAFIEC_ROUEN_WEBNET_POEC_201805

Nom : Prénom :

Thèmes traités :

1. Comprendre le fonctionnement de GIT et mettre en place un dépôt pour travailler en équipe
2. Algorithmique : analyser un problème et trouver un moyen de le résoudre
3. SQL : Savoir créer une base de données, ajouter et modifier des tables (SQL Server)
4. SQL : Ecrire des requêtes SQL (simples et avec jointures) et des procédures stockées
5. C# : Utiliser ADO.NET et Entity Framework pour travailler avec une base de données

Analysez les thèmes que vous avez acquis/compris : (utiliser des feuilles annexées)

Comprendre le fonctionnement de GIT et mettre en place un dépôt pour travailler en équipe :

J'ai pu utiliser GIT, seul et avec mon binôme et cela a été très pratique.
GIT permet de créer un dossier (repository) local (sur la machine en cours d'utilisation), qui contient le projet sur lequel on travaille. Ce dossier peut ensuite être « pousser » dans le cloud (nous utilisons GitHub) dans un dossier distant et ainsi être disponible depuis n'importe quelle machine connectée à internet.
L'intérêt pour le travail en binôme est que l'on peut déterminer qui sont les contributeurs qui peuvent accéder à/modifier ce dossier distant et ainsi travailler à plusieurs sur le même projet. Il faut par contre gérer les conflits lorsque des modifications sont faites sur le même élément par 2 contributeurs différents.

L'autre intérêt, à titre personnel, est que désormais je peux travailler sur mes différents projets à la fois de chez moi et depuis la salle de formation.

Algorithmique : analyser un problème et trouver un moyen de le résoudre :

L'algorithmique est primordial avant de commencer son code.
Le code doit permettre de répondre à une problématique, il faut donc être capable de définir avec des phrases simples le déroulement de notre code, pas à pas, jusqu'à la résolution de la problématique.
Cela permet de décomposer son futur code en courtes phrases qui contiendront des verbes d'actions à l'infinitif (correspondant le plus souvent aux méthodes de notre programme) et des noms (correspondant le plus souvent aux variables de notre programme). Une fois l'algorithme écrit, l'écriture du code se fait plus facilement.

SQL : Savoir créer une base de données, ajouter et modifier des tables (SQL Server) :

J'ai compris que le SQL est un langage permettant d'écrire des requêtes afin de communiquer avec une base de données. Les bases de données permettent de stocker et de structurer des quantités importantes d'informations à l'intérieur de tables. Elles sont hébergées sur des serveurs locaux ou distants. J'ai appris à utiliser SQL Server Management Studio pour créer un serveur local. Je peux me connecter au serveur et créer une base de données dans laquelle je sais créer des tables, les modifier ou les supprimer grâce aux requêtes suivantes : CREATE TABLE, ALTER TABLE, DROP TABLE. J'ai

compris la notion les notions de clés primaires, le plus souvent auto-incrémentées, et de clés secondaires, permettant de faire des jointures entre les tables.

SQL : Ecrire des requêtes SQL (simples et avec jointures) et des procédures stockées :

Je sais créer, lire, modifier ou supprimer une table présente dans une base de données. Que ce soit en utilisant des requêtes simples de type CREATE, SELECT, UPDATE, DELETE ou graphiquement en utilisant les outils de SQL Server Management Studio. Je sais effectuer des jointures en utilisant la méthode JOIN au sein d'une requête, en indiquant qu'elle est la colonne (possédant une clé secondaire) de la seconde table qui correspond à une colonne de ma première table. Cela permet de récupérer des données présentent sur différentes tables et de les agencer sous la forme d'un tableau unique.

Je n'ai malheureusement pas pu essayer de créer ou manipuler des procédures stockées car nous avons eu le temps d'à peine les citer.

C# : Utiliser ADO.NET et Entity Framework pour travailler avec une base de données :

J'ai compris qu'ADO.NET est une bibliothèque de classes permettant d'interagir avec une base de données relationnelle. Grace à ADO.NET on peut, au sein de notre application développée en C#, ouvrir une connexion vers une base de données et venir récupérer, ajouter, modifier ou supprimer ces données.

De plus le framework Entity, une fois configuré pour que nos classes métiers correspondent aux différentes tables de notre base de données, permet de s'affranchir de l'écriture de beaucoup de lignes de code pour arriver à interagir avec cette base de données. Cela permet un gain de temps considérable et rend aussi l'interaction avec la base de données plus intuitive puisque l'on retrouve une syntaxe « objet » avec laquelle je commence à être à l'aise.

Analysez les difficultés que vous avez rencontrées :

GIT :

Le fonctionnement de GIT n'est pas encore totalement intuitif lorsque l'on travaille à plusieurs sur le même projet. J'ai du mal à savoir à quel moment « tirer » les modifications faites par son binôme alors que je modifie moi-même une autre partie du document. Je crois qu'il faut d'abord que moi-même je pousse mes modifications et que je gère les conflits s'ils existent avant de pouvoir récupérer les modifications de mon binôme. Je vais m'entraîner sur ce point lors du prochain projet.

Algorithmique :

Je suis capable d'écrire l'algorithme de mon futur code mais je n'arrive pas encore à envisager l'ensemble des solutions possibles à ma problématique et ne peux donc pas « choisir » le meilleur algorithme (en termes de temps ou de ressource).

Je vais me renseigner sur les différentes approches à avoir quand on veut résoudre un problème.

J'aurais cependant préféré que l'on voit l'algorithmie avant ou en même temps que le C# et non après.

SQL :

J'ai vu les requêtes permettant de créer ou modifier les bases de données et les tables qu'elles contiennent mais le fait de pouvoir le faire graphiquement sans passer par des requêtes textuelles a tendance à me faire oublier la syntaxe exacte de ces dernières.

Je sais cependant que je pourrai aller chercher sur internet les syntaxes exactes si un jour j'ai besoin de m'en servir.

Quand une requête devient un peu compliquée et implique plusieurs jointures ou conditions, j'ai tendance à m'embrouiller dans l'ordre que je dois respecter pour écrire tous les paramètres. Encore une fois Internet et les supports de la formation pourront m'aider à retrouver la solution.

En ce qui concerne les procédures stockées il va falloir que j'aille voir dans le support de cours et/ou sur internet pour pouvoir les comprendre et les utiliser.

ADO.NET et Entity Framework

Comme pour SQL au sein de SQL Server Management Studio, l'écriture des requêtes avec ADO.NET reste assez fastidieuse et je dois me référer au support de cours pour être syntaxiquement irréprochable.

Cette difficulté est majoritairement évitée avec l'utilisation d'Entity Framework, cependant c'est la configuration du Framework qui pose problème. J'utilise la méthode « Code First » et ne rien oublié lors de la configuration de mes classes métiers (la définition des clés secondaires, ou le « renommage » des colonnes par exemple) est compliqué. Encore une fois, au besoin je sais où trouver les infos, et je pense que la pratique me permettra d'acquérir tous les automatismes.

Relations avec le groupe :

Je m'entends toujours très bien avec l'ensemble des participants de la formation, que ce soit professionnellement ou humainement. Je fais en sorte de les aider quand je le peux, et j'essaie de reformuler/paraphraser, avec des mots plus compréhensibles pour les novices que nous sommes, les explications du formateur quand cela est nécessaire. Cela me permet aussi de me rendre compte si j'ai bien compris ce que l'on nous enseigne.

Suggestions – Attentes :

Plus de temps et surtout de la pratique sur les procédures stockées.

Travailler de soi-même le soir après une longue journée de formation est souvent compliqué mais, je le vois, est nécessaire. Il est parfois difficile de savoir quoi faire pour s'entraîner et progresser. Je suggère que régulièrement (disons un jour sur deux) les formateurs nous donnent de petits exercices assez courts, à faire chez soi, en relation avec le cours de la journée.