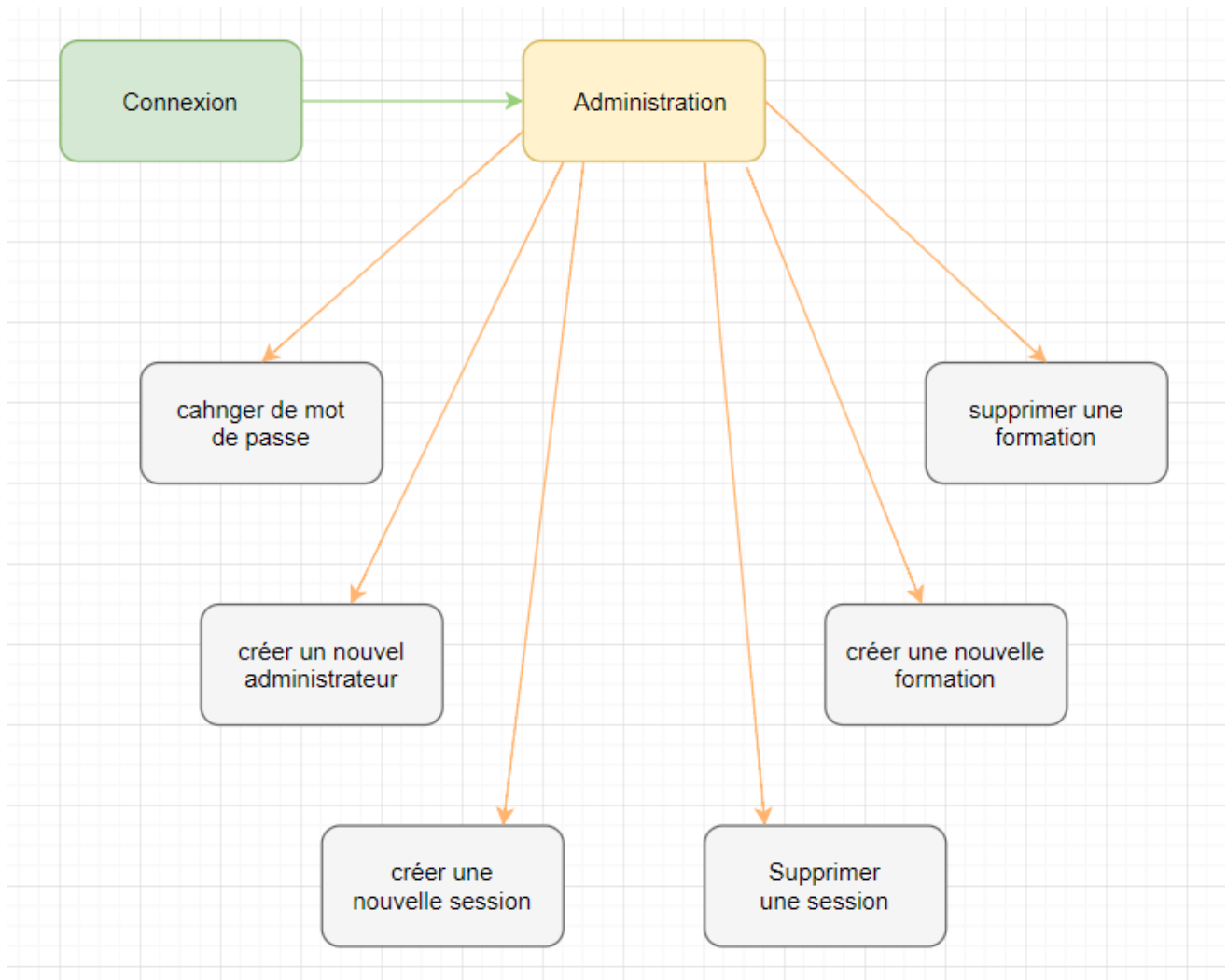


Documentation Technique Application Gestion de sondages

ARBORESCENCE



BASE DE DONNEES

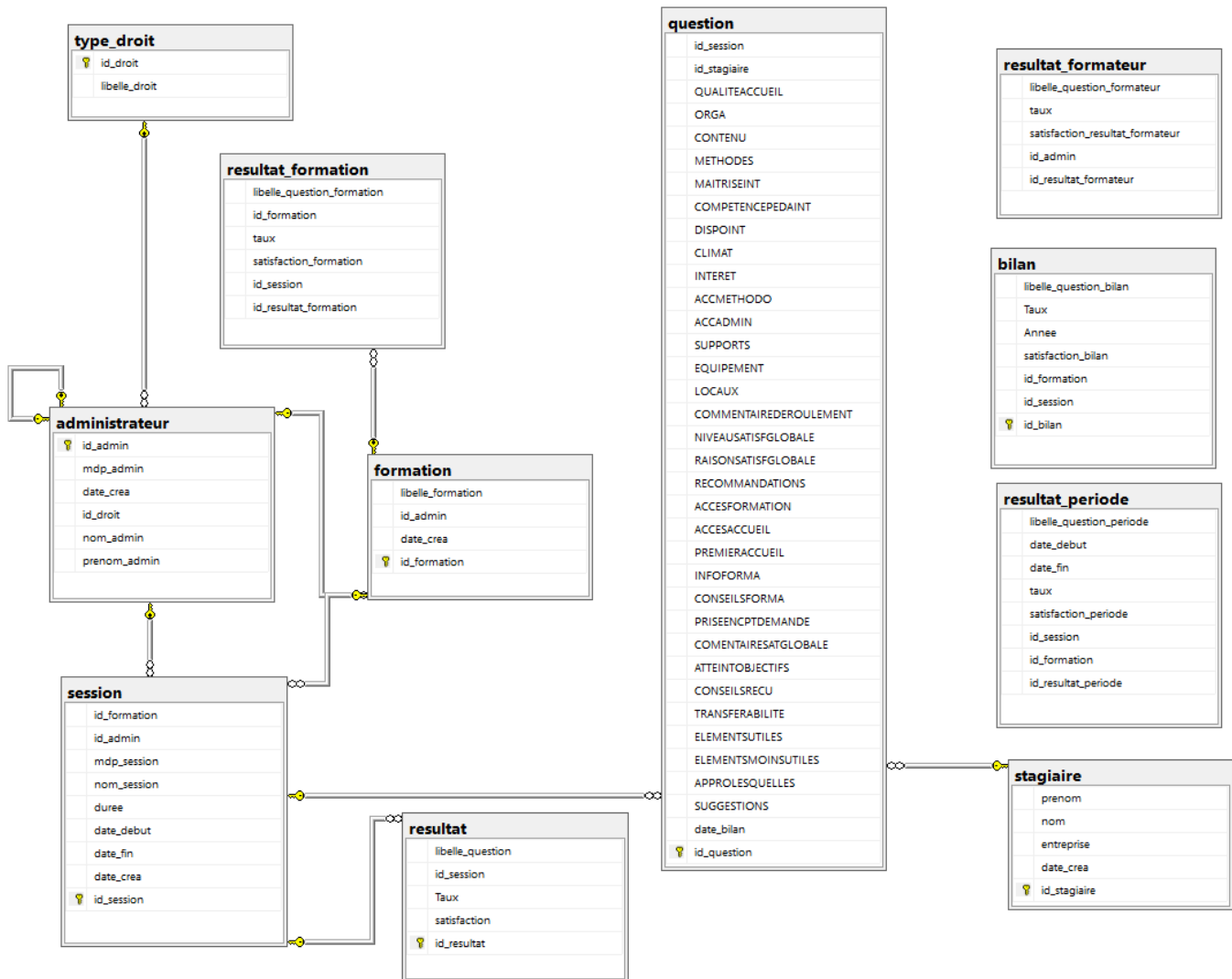


Table Administrateur :

Nom	type	null	clef
Id_admin	Char(32)	non	primaire
Mdp_admin	Char(32)	Non	
Date_crea	Date	Non	
Id_droit	Int(1)	Non	

Id_admin : correspond à l'identifiant qui permettra à l'utilisateur de se connecter, cet identifiant est unique

Mdp_admin : correspond au mot de passe qui permettra à l'administrateur de se connecter

Date_crea : correspond à la date de création de l'administrateur

Id_droit : correspond à l'identifiant du niveau de droit attribué à l'administrateur

Table Bilan :

Nom	type	null	clef	extra
id_bilan	int(11)	non	primaire	auto incrément
libelle_question_bilan	varchar(100)	non		
taux	float	non		
annee	int(4)	non		
satisfaction_bilan	enum(EPP,S,PS,TS,Demande,Proposition,Impose,Oui,Non)	non		
id_formation	smalint(1)	non	étrangère	
id_session	int(2)	non	étrangère	

Id_bilan : correspond à l'identifiant du bilan

Libelle_question_bilan : correspond au libelle de la question dont on va récupérer le résultat

Taux : correspond au taux de satisfaction relevé pour la question

Annee : correspond à l'année sur laquelle porte le bilan

Satisfaction_bilan : correspond au libelle de satisfaction de la question

Id_formation : correspond à l'identifiant de la formation sur laquelle porte le bilan

Id_session : correspond à l'identifiant de la session sur laquelle porte le bilan

Table formation :

Nom	Type	Null	Clef	Extra
Id_formation	Smallint(1)	Non	primaire	Auto incrément
Libelle_formation	Varchar(50)	Non		
Id_admin	Char(32)	Non	Etrangère	
Date_crea	date	non		

id_formation : identifiant de la formation

libelle_formation : libelle de la formation

id_admin : identifiant de l'administrateur qui a créé la formation

date_crea : date de création de la formation

Table question :

nom	type	null	clef	extra
Id_question	smallint(1)	Non	Primaire	Auto incrément
Id_session	int(2)	Non	Etrangère	
Id_stagiaire	smallint(6)	Non	Etrangère	
QUALITEACCUEIL	enum(EPS,S,PS,TS)	Non		
ORGA	enum(EPS,S,PS,TS)	Non		
CONTENU	enum(EPS,S,PS,TS)	Non		
METHODES	enum(EPS,S,PS,TS)	Non		
MAITRISEINT	enum(EPS,S,PS,TS)	Non		
COMPETENCESPEDAINT	enum(EPS,S,PS,TS)	Non		
DISPOINT	enum(EPS,S,PS,TS)	Non		
CLIMAT	enum(EPS,S,PS,TS)	Non		
INTERET	enum(EPS,S,PS,TS)	Oui		
ACCMETHODO	enum(EPS,S,PS,TS)	Non		
ACCADMIN	enum(EPS,S,PS,TS)	oui		
SUPPORTS	enum(EPS,S,PS,TS)	Non		
EQUIPEMENT	enum(EPS,S,PS,TS)	Non		
LOCAUX	enum(EPS,S,PS,TS)	Non		
COMMENTAIREDEROULEMENT	varchar(500)	Oui		
NIVEAUSATISFAGLOBALE	enum(EPS,S,PS,TS)	Non		
RAISONSATISFGLOBALE	varchar(500)	Oui		
RECOMMANDATION	enum(Oui,Non)	Non		
ACCESFORMATION	enum(Demande,Proposition,Impose)	Non		
ACCESACCUEIL	enum(EPS,S,PS,TS)	Oui		
PREMIERACCUEIL	enum(EPS,S,PS,TS)	Oui		
INFOFORMA	enum(EPS,S,PS,TS)	Oui		
CONSEILSFORMA	enum(EPS,S,PS,TS)	Oui		
PERISEENCPTDEMANDE	enum(EPS,S,PS,TS)	Oui		
COMMENTAIRESATGLOBALE	enum(EPS,S,PS,TS)	Oui		

ATTEINTOBJECTIFS	enum(EPS,S,PS,TS)	Non		
CONSEILSRECU	enum(EPS,S,PS,TS)	Oui		
TRANSFERABILITE	enum(EPS,S,PS,TS)	Non		
ELEMENTSUTILES	Varchar(500)	Oui		
ELEMENTSMOINSUTILES	Varchar(500)	Oui		
APPROLESQUELLES	Varchar(500)	Oui		
SUGGESTIONS	Varchar(500)	Oui		
Date_bilan	Date	Non		

id_question : identifiant de la question

Id_session : identifiant de la session a laquelle appartient le stagiaire qui remplit le bilan

Id_stagiaire : identifiant du stagiaire qui remplit le bilan

Pour les champs Enum(EPS,S,PS,TS) :

- EPS : en partie satisfait(e)
- S : satisfait(e)
- PS : pas satisfait(e)
- S : satisfait(e)

QUALITEACCUEIL : résultat de la question « qualité de l'accueil au démarrage de la séance de formation »

ORGA : résultat de la question « Organisation de la formation »

CONTENU : résultat de la question « contenu de la formation »

METHODES : résultat de la question « méthodes et activités pédagogiques »

MAITRISEINT : résultat de la question « maitrise du sujet » de la catégorie « qualité des intervenants »

COMPETENCEPEDAINT : résultat de la question « compétences pédagogiques » de la catégorie « qualité des intervenants »

DISPOINT : résultat de la question « disponibilité et écoute » de la catégorie « qualité des intervenants »

CLIMAT : résultat de la question « climat de travail et d'échange avec les autres stagiaires »

INTERET : résultat de la question « le cas échéant, intérêt de la phase d'immersion au sein d'une séance de formation »

ACCMETHODO : résultat de la question « Qualité de l'accompagnement méthodologique et/ou pédagogiques »

ACCADMIN : résultat de la question « Qualité de l'accompagnement administratif (contractualisation, dossier de financement et/ou de rémunération)

SUPPORTS : résultat de la question « qualité des supports pédagogiques utilisés et/ou remis »

EQUIPEMENT : résultat de la question « équipements techniques et informatiques »

LOCAUX : résultat de la question « locaux »

COMMENTAIREDEROULEMENT : résultat de la question « commentaires » de la catégorie « Déroulement »

NIVEAUSATISFGLOBALE : résultat de la question « satisfaction globale de la formation »

RAISONSATISFGLOBALE : résultat de la question « pourquoi ? » pour la satisfaction globale de la formation

Pour le champ Enum(Oui,Non) :

- Oui : Oui
- Non : Non

RECOMMANDATIONS : résultat de la question « recommanderiez-vous la formation à d'autres personnes »

Pour le champ Enum(Demande,Proposition,Impose) :

- Demande : c'est une demande de ma part
- Proposition : c'est une proposition qui m'a été faite
- Impose : on me l'a imposé

ACCESFORMATION : résultat de la question « comment avez-vous été conduit à suivre cette formation »

ACCESACCUEIL : résultat de la question « accès au lieu d'accueil »

PREMIERACCUEIL : résultat de la question « premier accueil (téléphone, mail, visite)

INFOFORMA : résultat de la question « informations données sur la formation »

CONSEILSFORMA : résultat de la question « conseils sur les formations et financements »

PRISEENCPTDEMANDE : résultat de la question « prise en compte de ma demande »

COMMENTAIRESATGLOBALE : résultat de la question « commentaires » de la catégorie « satisfaction globale »

ATTEINTOBJECTIFS : résultat de la question « atteinte de vos objectifs »

CONSEILSRECU : résultat de la question « le cas échéant, conseils reçus pour la suite »

TRANSFERABILITE : résultat de la question « transférabilité du contenu de la formation en situation professionnelle »

ELEMENTSUTILES : résultat de la question « quels sont les éléments de la formation qui vous ont paru les plus pertinents et les plus utiles »

ELEMENTSMOINSUTILES : résultat de la question « quels sont les éléments de la formation qui vous ont le moins intéressés »

APPROLESQUELLES : résultat de la question « souhaiteriez-vous approfondir certaines thématiques ? si oui, lesquelles ? »

SUGGESTIONS : résultat de la question « avez-vous des suggestions pour améliorer la qualité de nos formations et notre organisation »

Date_bilan : date de remplissage du bilan

Table Resultat (par sessions) :

Nom	type	null	clef	extra
Id_resultat	Int(1)	Non	Primaire	Auto incrément
Libelle_question	Varchar(100)	Non		
Id_session	Int(2)	Non	étrangère	
Taux	Float	Non		
satisfaction	enum(EPP,S,PS,TS,Demande,Proposition,Impose,Oui,Non)	Non		

Id_résultat : identifiant de la ligne de résultat

Libelle_question : libelle de la question correspondant au résultat

Id_session : libelle de la session correspondant au résultat

Taux : pourcentage de satisfaction pour la question

Satisfaction : libelle de satisfaction correspondant au résultat

Table Resultat_formateur :

Nom	type	null	clef	extra
Id_resultat_formateur	Int(1)	Non	Primaire	Auto incrément
Libelle_question_formateur	Varchar(50)	Non		
Taux	Float	Non		
Satisfaction_resultat_formateur	enum(EPP,S,PS,TS,Demande,Proposition,Impose,Oui,Non)	Non		
Id_admin	Char(32)	Non	étrangère	

Id_resultat_formateur : identifiant de la ligne de résultat

Libelle_question_formateur: libelle de la question correspondant au résultat

Taux : pourcentage de satisfaction pour la question

Satisfaction_resultat_formateur : libelle de satisfaction correspondant au résultat

Id_admin : identifiant de l'administrateur (formateur) qui correspond au résultat

Table Resultat formation :

Nom	type	null	clef	extra
Id_resultat_formation	Int(1)	Non	Primaire	Auto incrément
Libelle_question_formation	Varchar(50)	Non		
Id_formation	Smallint(1)	Non	étrangère	
Taux	Float	Non		
Satisfaction_formation	enum(EPP,S,PS,TS,Demande, Proposition,Impose,Oui,Non)	Non		
Id_session	Int(2)	Oui	étrangère	

Id_resultat_formation : identifiant de la ligne de résultat

Libelle_question_formation: libelle de la question correspondant au résultat

Id_formation : identifiant de la formation correspondant au résultat

Taux : pourcentage de satisfaction pour la question

Satisfaction_formation : libelle de satisfaction correspondant au résultat

Id_session: identifiant de la session correspondante au résultat

Table Resultat_période :

Nom	type	null	clef	extra
Id_resultat_période	Int(11)	Non	Primaire	Auto incrément
Libelle_question_période	Varchar(50)	Non		
Date_debut	date	Non		
Date_fin	date	Non		
Taux	Float	Non		
Satisfaction_période	enum(EPP,S,PS,TS,Demande, Proposition,Impose,Oui,Non)	Non		
Id_session	Int(2)	Oui	étrangère	
Id_formation	Smallint(1)	Oui	étrangère	

Id_resultat_formation : identifiant de la ligne de résultat

Libelle_question_formation: libelle de la question correspondant au résultat

Date_debut : date de débit de la période

Date_fin : date de fin de la periode

Taux : pourcentage de satisfaction pour la question

Satisfaction_formation : libelle de satisfaction correspondant au résultat

Id_session: identifiant de la session correspondante au résultat

Id_formation : identifiant de la formation correspondant au résultat

Table Session :

Nom	type	null	clef	extra
Id_session	Int(2)	Non	Primaire	Auto incrément
Id_formation	Smallint(1)	Non	Etrangère	
Id_admin	Char(32)	Non	Etrangère	
Mdp_session	Char(32)	Non		
Nom_session	Varchar(50)	Non		
Duree	enum('Moins50h','50a100','Plus100h')	Non		
Date_debut	Date	Non		
Date_fin	Date	Non		
Date_crea	Date	Non		

Id_session : identifiant de la session

Id_formation : identifiant de la session a laquelle correspond la session

Id_admin : identifiant de l'administrateur ayant créé la session

Mdp_session : mot de passe de la session

Nom_session : nom de la session

Duree : durée de la session

Date_debut : date de début de la session

Date_fin : date de fin de la session

Date_crea : date de création de l'occurrence

Table stagiaire :

Nom	type	null	clef	extra
Id_stagiaire	Smallint(1)	Non	primaire	Auto incrément

Prenom	Char(30)	Non		
Nom	Char(40)	Non		
Entreprise	Varchar(50)	Non		
Date_crea	Date	Non		

Id_stagiaire : identifiant du stagiaire

Prenom : prénom du stagiaire

Nom : nom du stagiaire

Entreprise : entreprise du stagiaire

Date_crea : date de création du stagiaire dans la base

Table type_droit :

Nom	type	null	clef
Id_droit	Int(1)	Non	Primaire
Libelle_droit	Varchar(50)	Non	

Id_droit : identifiant du niveau de droit

Libelle_droit : libelle du niveau de droit

FONCTIONS

Administrateur.ConfirmationMdp()

Description :

```
Public static bool Confirmationmdp(string mdp1, string mdp2)
```

Permet de vérifier si les mots de passe entrés sont les mêmes.

Liste de paramètres :

Mdp1 : premier mot de passe à comparer

Mdp2 : second mot de passe à comparer

Valeurs de retour :

Retourne un booléen vrai si les mots de passe en entrée sont identiques, faux dans le cas contraire.

Exemple :

```
Administrateur.Confirmationmdp(«bonjour»,«bonsoir»);
```

Retournera faux.

Administrateur.OuvertureConnexion()

Description :

```
Public static void OuvertureConnexion()
```

Permet d'ouvrir la connexion à la base de données.

Exemple-:

```
Administrateur.OuvertureConnexion();
```

Administrateur.FermetureConnexion()

Description :

```
Public static void FermetureConnexion()
```

Permet de fermer la connexion à la base de données.

Exemple-:

```
Administrateur.FermetureConnexion();
```

Administrateur.AdmExisteDeja()

Description :

```
Public static bool AdmExisteDeja(string id)
```

Permet de savoir si l'administrateur existe déjà.

Liste de paramètres :

Id : identifiant de l'administrateur à vérifier

Valeurs de retour :

Retourne un booléen vrai si l'administrateur existe déjà, faux dans le cas contraire.

Exemple :

```
Administrateur.Confirmationmdp(«lmaurice»);
```

Administrateur.CreationDateJour()

Description :

```
Public static void CreationDateJour()
```

Permet d'avoir la date du jour de façon à pouvoir la rentrer plus tard dans la base de données.

Valeurs de retour :

Retourne la date du jour en un string au format qui pourra être entré dans la base de données.

Exemple :

```
Administrateur.CreationDateJour();
```

Administrateur.SessionExisteDeja()

Description :

```
Public static bool SessionExisteDeja(string id)
```

Permet de savoir si la session existe déjà.

Liste de paramètres :

Id : identifiant de la session à vérifier

Valeurs de retour :

Retourne un booléen vrai si la session existe déjà, faux dans le cas contraire.

Exemple :

```
Administrateur.SessionExisteDeja(«session 1 module 1»);
```

Administrateur.ModificationDateBdd()

Description :

```
Public static string ModificationDateBdd(Datetime date)
```

Permet de convertir un DateTime en string qui aura le format de date qui pourra être entré dans la base de données.

Liste de paramètres :

date : date à convertir

Valeurs de retour :

Retourne un string au format de la base de données.

Exemple :

```
DateTime Date = new DateTime(2018,03,05) ;  
Administrateur.SessionExisteDeja(Date);
```

Administrateur.VerificationDates()

Description :

```
Public static bool VerificationDates(Datetime date1, Datetime date2)
```

Permet de savoir si la date1 est supérieure à la date2.

Liste de paramètres :

Date1 : première date.

Date2 : deuxième date

Valeurs de retour :

Retourne un booléen a vrai si la date1 est supérieure à la date1, faux dans le cas contraire.

Exemple :

```
DateTime Date1 = new DateTime(2018,03,05) ;
```

```
DateTime Date2 = new DateTime(2018,03,01) ;
```

```
Administrateur.VerificationDates(Date1,Date2);
```

Retournera faux

Administrateur.VerifFormaExisteDeja()

Description :

```
Public static bool VerifFormaExisteDeja(string nomforma)
```

Permet de vérifier si la formation existe déjà.

Liste de paramètres :

nomforma : nom de la formation a tester

Valeurs de retour :

Retourne un booléen a vrai si la formation existe déjà, faux dans le cas contraire.

Exemple :

```
Administrateur.VerifFormaExisteDeja(«formation1»);
```

FM_Connexion.ConnexionAdm()

Description :

```
private void ConnexionAdm(string idsaisi, string mdpsaisi)
```

Permet à l'administrateur de se connecter.

Liste de paramètres :

Id : identifiant de l'administrateur

Mdpsaisi : mot de passe de l'administrateur

Exemple :

```
ConnexionAdm(«lmaurice»,«mdp»);
```

FM_Connexion.VerifChampsRemplisCo ()

Description :

```
private bool VerifChampsRemplisCo(string id, string mdp)
```

Permet de vérifier si tous les champs du formulaire sont remplis.

Liste de paramètres :

Id : champs id

Mdp : champ mot de passe

Valeurs de retour :

Retourne un booléen a vrai si les champs sont tous remplis, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisCo(«lmaurice», «Mdp»);
```

Retourne Vrai

FM_ChangerMdp.ChangerMdp ()

Description :

```
private void ChangerMdp()
```

Permet à l'administrateur de changer son mot de passe.

Exemple :

```
ChangerMdp();
```

FM_ChangerMdp.VerifChampsRemplisCM ()

Description :

```
private bool VerifChampsRemplisCM(string mdp1, string mdp2)
```

Permet de vérifier si tous les champs du formulaire sont remplis.

Liste de paramètres :

Mdp1: champ mot de passe 1
Mdp2 : champ mot de passe 2

Valeurs de retour :

Retourne un booléen a vrai si les champs sont tous remplis, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisCM(«mdp», «Mdp»);  
Retourne Vrai
```

FM_CreationAdmin.CreationAdm ()

Description :

```
private void CreationAdm(int nvdroit, string id,string mdp ,string mdp2  
,string nom,string prenom)
```

Permet à l'administrateur de créer un nouvel administrateur.

Liste de paramètres :

Nvdroit : niveau de droit à accorder à l'administrateur

Id : identifiant de l'administrateur à créer

Mdp : mot de passe de l'administrateur à créer

Mdp2 : confirmation du mot de passe de l'administrateur à créer

Nom : nom de l'administrateur à créer

Prénom : prénom de l'administrateur à créer

Exemple :

```
CreationAdm(1,«lmaurice», «Mdp», «Mdp», «maurice», «léa» );
```

FM_CreationAdmin.VerifChampsRemplisCA ()

Description :

```
private bool VerifChampsRemplisCA(string nvdroit, string id,string mdp  
,string mdp2 ,string nom,string prenom)
```

Permet de vérifier si tous les champs du formulaire sont remplis.

Liste de paramètres :

Nvdroit : niveau de droit à accorder à l'administrateur

Id : identifiant de l'administrateur à créer

Mdp : mot de passe de l'administrateur à créer

Mdp2 : confirmation du mot de passe de l'administrateur à créer

Nom : nom de l'administrateur à créer

Prénom : prénom de l'administrateur à créer

Valeurs de retour :

Retourne un booléen a vrai si les champs sont tous remplis, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisCA(«1»,«lmaurice», «Mdp», «Mdp», «maurice», «léa» );
```

Retourne Vrai

FM_CreationSession.VerifChampsRemplisCS ()

Description :

```
private bool VerifChampsRemplisCS(string idformation, string  
duree,string nom,string datedebut ,string datefin,string mdp1,string mdp2)
```

Permet de vérifier si tous les champs du formulaire sont remplis.

Liste de paramètres :

Idformation : identifiant de la formation correspondante à la session
duree : durée de la session
nom : nom de la session
datedebut : date de début de la session
datefin : date de fin de la session
mdp : mot de passe
mdp2 : confirmation de mot de passe

Valeurs de retour :

Retourne un booléen a vrai si les champs sont tous remplis, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisCS(«1»,«50a100», «session1», «2018-02-02», «2018-02-20»,  
«mdp», «mdp» );  
Retourne Vrai
```

FM_CreationSession.CreationSession ()

Description :

```
private void CreationSession(string idformation, string duree,string  
nom,string datedebut ,string datefin,string mdp)
```

Crée une nouvelle session dans la base de données.

Liste de paramètres :

Idformation : identifiant de la formation correspondante à la session
duree : durée de la session
nom : nom de la session
datedebut : date de début de la session
datefin : date de fin de la session
mdp : mot de passe
mdp2 : confirmation de mot de passe

Exemple :

```
CreationSession(«1»,«50a100», «session1», «2018-02-02», «2018-02-20»,  
«mdp», «mdp» );  
Retourne Vrai
```

FM_CreationFormation.VerifChampsRemplisCF ()

Description :

```
private void CreationSession(string idformation)
```

Vérifie si le champ est rempli.

Liste de paramètres :

Idformation : identifiant de la formation

Valeurs de retour :

Retourne un booléen a vrai si le champ est rempli, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisCF(«1»);  
Retourne Vrai
```

FM_CreationFormation.VerifChampsRemplisSS ()

Description :

```
private void VerifChampsRemplisSS(string session)
```

Vérifie si le champ est rempli.

Liste de paramètres :

Idformation : identifiant de la formation

Valeurs de retour :

Retourne un booléen a vrai si le champ est rempli, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisSS(«session»);  
Retourne Vrai
```

FM_CreationFormation.VerifChampsRemplisSF ()

Description :

```
private void VerifChampsRemplisSS(string formation)
```

Vérifie si le champ est rempli.

Liste de paramètres :

Idformation : identifiant de la formation

Valeurs de retour :

Retourne un booléen a vrai si le champ est rempli, faux dans le cas contraire.

Exemple :

```
VerifChampsRemplisSF(«1»);  
Retourne Vrai
```
