

Secteur Tertiaire Informatique  
Filière « Etude et développement »

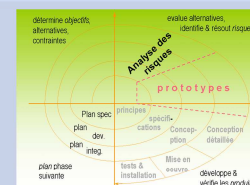
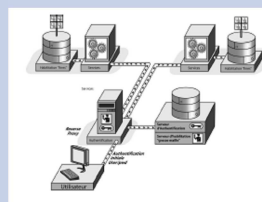
Mettre en place une base de données

**Cas GESPERSO**  
**Création de la base de données – SQL Server**

Apprentissage

Mise en situation

Evaluation





## SOMMAIRE

Sommaire .....	3
1. Rappels.....	5
1.1 Le dictionnaire des données .....	5
1.2 Le modèle conceptuel des données.....	6
1.3 Le modèle physique pour SQL Server.....	7
1.4 Les contraintes.....	8
2. Création de la base de données.....	9
2.1 Activité 1 - Création de la base de données .....	9
2.2 Activité 2 – Test de la structure de la base .....	10
2.3 Activité 3 - Création d'utilisateurs .....	11
2.4 Activité 4 - Création de rôles .....	13
2.5 Activité 5 - Sauvegarde et Restauration de la base de test .....	14

## Préambule

Etude de cas à réaliser dans le cadre de l'apprentissage de la création d'une base de données relationnelle à l'aide d'un outil de modélisation puis de son administration de base (droits d'accès et sauvegarde/restauration).

## Objectifs

Mettre en œuvre les instructions SQL de création d'une base de données relationnelle et utiliser des outils du SGBD SQL Server.

## Méthodologie

Après la prise en compte du cahier des charges et des modèles de données, exposés dans le document P-cas-gesperso.pdf, réaliser les étapes proposées dans les différentes activités.

## Ressources

- Base de données SQL Server (2012), installée.
- Utilitaire (EDI) SQL Server Management Studio.
- Les scripts LDD SQL du cas GESPERSO générés par votre outil de modélisation Merise.

## 1. RAPPELS

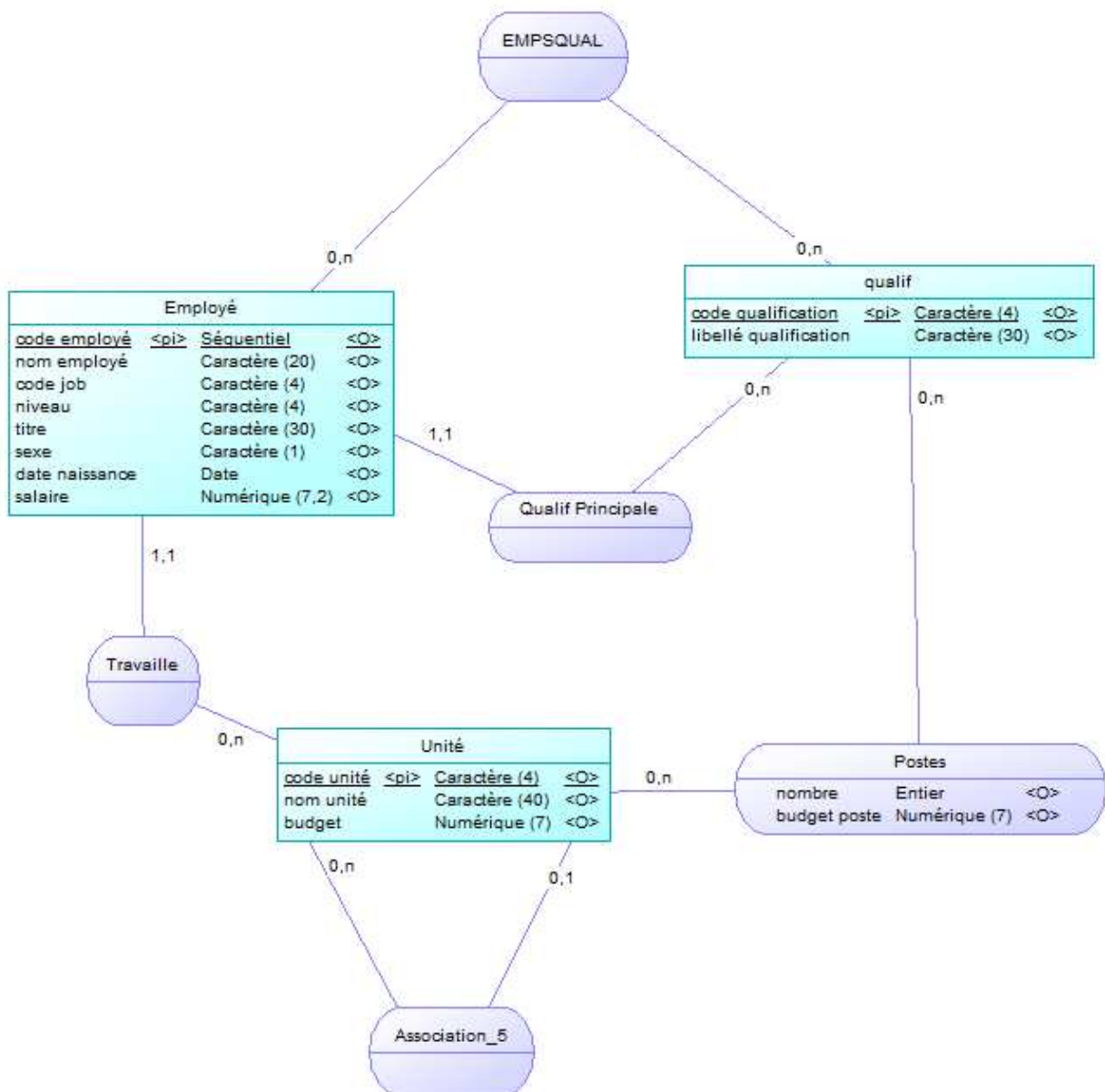
La société DUBILAN et FILS, spécialisée dans la production industrielle, souhaite gérer son personnel à partir des règles de gestion décrites ci-dessous :

- ✓ Les personnels sont enregistrés avec une qualification principale mais peuvent avoir une ou plusieurs qualifications secondaires.
- ✓ Ces personnels sont affectés à des unités de production. Ces unités de production dépendent d'une unité principale dite "mère".
- ✓ A chaque unité sont associées des qualifications qui définissent des postes de travail. Les qualifications peuvent être présentes ou non dans chacune des unités.
- ✓ Chaque unité dispose d'un budget global de fonctionnement, réparti sur l'ensemble des postes de cette unité.

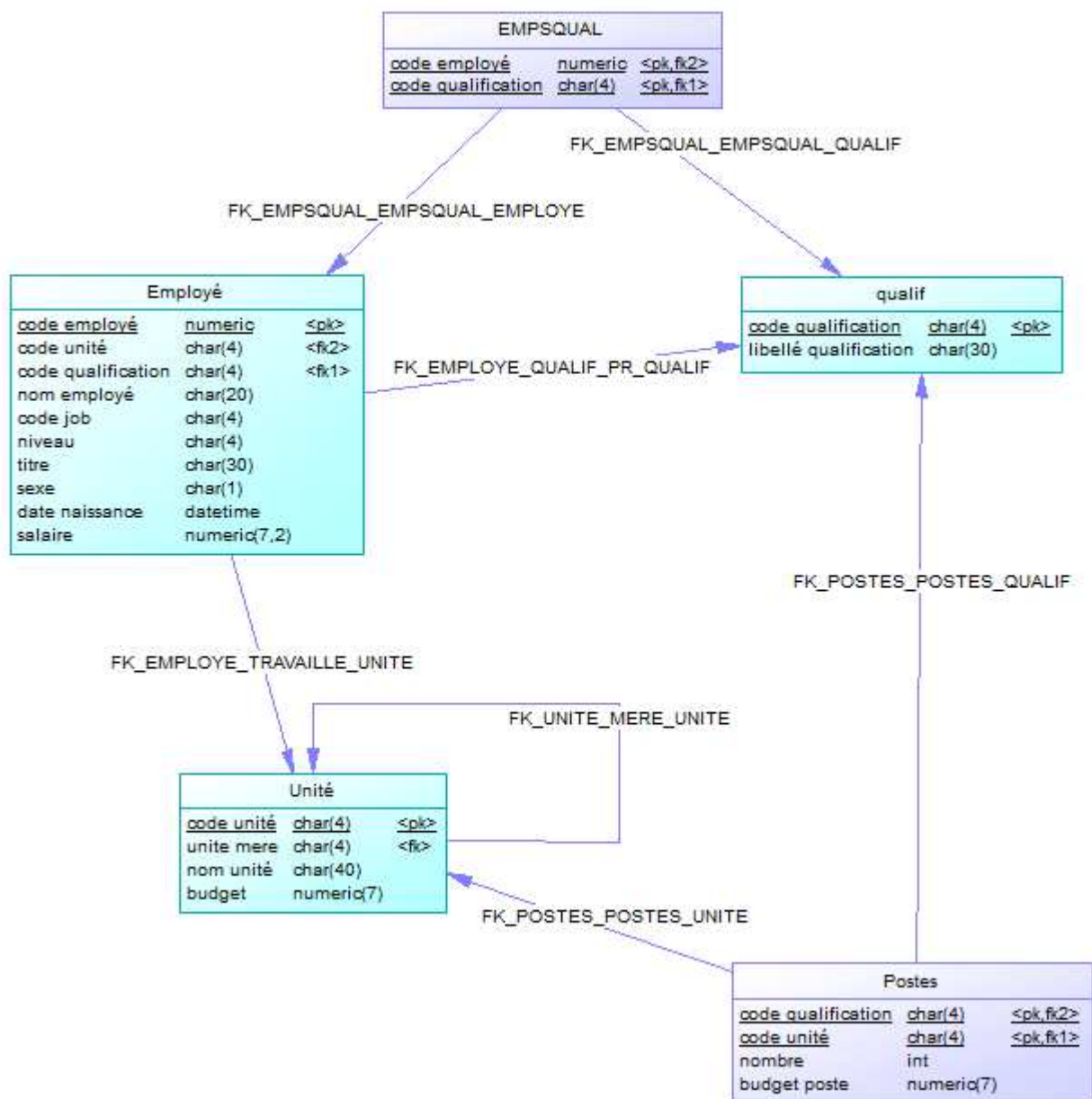
### 1.1 LE DICTIONNAIRE DES DONNEES

Libellé	Code	Type et longueur	Contraintes
Budget	BUDGET	NUM 7,0	
Budget Poste	BUDGETPOSTE	NUM 7,0	
Code Employé	EMPCOD	CAR 5	
Code Job	JOB CODE	CAR 4	
Code Qualification	CODQUAL	CAR 4	
Code Unité	UNITCOD	CAR 4	
Date Naissance	DATNAISS	DATE	
Libelle Qualification	QUALIBEL	CAR 30	
Niveau	NIVEAU	CAR 4	
Nom Employé	NOM	CAR 20	
Nom Unité	UNITE	CAR 40	
Nombre	NOMBRE	ENTIER	
Salaire	SALAIRE	NUM 7,2	
Sexe	SEXE	CAR 1	'F' ou 'M'
Titre	TITRE	CAR 30	

## 1.2 LE MODELE CONCEPTUEL DES DONNEES



### 1.3 LE MODELE PHYSIQUE POUR SQL SERVER



## 1.4 LES CONTRAINTES

- **Table EMPLOYE**
  - Tous les champs sont obligatoires.
  - Le code sexe est égal à 'F' ou 'M'.
  - Le code employé est un champ compteur auto incrémenté de 1.
  - Le code unité doit exister dans la table UNITE
  - Le code qualification doit exister dans la table QUALIF.
  
- **Table UNITE**
  - Tous les champs obligatoires sauf Unité Mère
  
- **Table QUALIF**
  - Tous les champs sont obligatoires
  
- **Table POSTES**
  - Tous les champs sont obligatoires
  - Le nombre est compris entre 1 et 100
  - Le budget doit être positif et différent de 0.
  - Le code unité doit exister dans la table UNITE
  - Le code qualification doit exister dans la table QUALIF
  
- **Table EMPSQUAL**
  - Tous les champs sont obligatoires.
  - Le code employé doit exister dans la table EMPLOYE
  - Le code qualification doit exister dans la table QUALIF



## 2. CREATION DE LA BASE DE DONNEES

### 2.1 ACTIVITE 1 - CREATION DE LA BASE DE DONNEES

Certaines tâches peuvent être réalisées au moyen du langage **LDD**, mais aussi grâce aux **assistants graphiques** : tester les deux manières de faire.

#### Sous SQL Server Management Studio :

##### Etape 1 :

1. Créez une connexion à la base de données de l'instance SQL Server en cours avec votre compte utilisateur.
2. Un script SQL généré par votre outil de modélisation Merise contient une description de la structure de la base. En cliquant sur l'icône **Ouvrir un fichier**, ouvrez ce fichier. Effectuez une relecture critique et affinez éventuellement le script généré. Exécutez le script dans SQL Server Management Studio.

*NB : Ce script généré par votre outil de modélisation ne devrait pas provoquer d'erreurs d'exécution. Si c'est le cas, vérifiez le SGBD et la version ciblés par votre outil Merise. Choisissez au besoin une cible plus 'générique' de manière à obtenir un script s'exécutant sans erreurs même s'il vous semble incomplet.*

##### Etape 2 :

1. A partir de l'inventaire des contraintes à implémenter, vérifiez leur bonne implémentation dans la structure de la base et complétez au besoin, à l'aide de l'outil d'administration SQL Server Management Studio.  
Pour les contraintes de clés étrangères, réfléchissez au cas par cas sur les actions à entreprendre en cas de suppression ou de mise à jour.
2. Créez un index sur la colonne NOM de la table EMPLOYE de manière à optimiser les recherches et tris futurs.

## 2.2 ACTIVITE 2 – TEST DE LA STRUCTURE DE LA BASE

### Etape 1 : insertion de lignes

Grâce à l'interface graphique SQL Server Management Studio ou aux instructions du langage DML (**INSERT**, **UPDATE**, **DELETE**), vous allez pouvoir ajouter, modifier et supprimer des lignes dans les tables de la base de données.

Mais auparavant, un temps de réflexion ....

La structure de la base de données vous impose de remplir les tables dans un ordre défini.

Quel est l'ordre à adopter ?

Et passez à l'acte (insérez des données), en proposant des données aussi bien valides que violant les contraintes d'intégrité. Vérifiez que les contraintes codées dans la structure de la base refusent les données erronées. Analysez les messages d'erreur. Rectifiez/complétez au besoin les contraintes déclaratives.

### Etape 2 : mise à jour de lignes

Modifiez de même des données avec l'ordre **UPDATE**. Testez aussi l'ordre **DELETE** sur différentes tables, notamment sur la table UNITE avec une unité absente et une présente sur les tables liées.

### Etape 3 : génération d'un script

Générez le script de votre base de données,

En pointant la base de données, **menu Tâches/Générer des scripts**. Ajustez les options nécessaires.

Analysez les écarts entre ce script et celui qui a été généré par votre outil de modélisation.

## 2.3 ACTIVITE 3 - CREATION D'UTILISATEURS

*Attention : Cette activité nécessite des droits de niveau Administrateur.*

SQL Server permet de gérer les droits d'accès avec sa propre base d'utilisateurs ou à partir des comptes Windows (notion de 'sécurités intégrées'). Pour les besoins courants, en particulier pour les formations, on adopte en général ce dernier mode d'authentification. Les consignes données ici correspondent à ce mode de 'sécurités intégrées'. En conséquence, il sera nécessaire de créer les utilisateurs au niveau de Windows et de basculer fréquemment entre les sessions d'utilisateurs ouvertes pour réaliser les manipulations.

Vous pouvez réaliser toutes les opérations de gestion des droits soit à travers des manipulations de SQL Server Management Studio, soit à l'aide des commandes SQL correspondantes, ou encore en alternant les 2. Pour les manipulations interactives, vous pouvez vous inspirer du document complémentaire **A-Acces-SQLServer2012.pdf** ; pour les commandes SQL, les documents **A-Securite-SQLServer-2008.pdf** et **A-sauvegards-restauration-SQLServer.pdf** vous fournissent des pistes et les syntaxes principales.

Vous allez créer et autoriser des utilisateurs à accéder à SQL Server et à la base de données **GesPerso**.

### Etape 1 : Création d'un utilisateur de votre serveur SQL Server :

Créez l'utilisateur Windows local '**User1**', qui ne soit pas Administrateur ('utilisateur standard', 'utilisateur avec pouvoirs'...).

En session Administrateur, à l'aide de manipulations interactives ou de commandes, sous SQL Server Management Studio, créez une nouvelle **Connexion** à votre serveur SQL Server pour cet utilisateur **User1** en précisant les paramètres suivants :

- Authentification : **Windows**
- Base de données par défaut : **GesPerso**
- Autres options : par défaut

Basculez en session Windows **User1** et testez une connexion avec cet utilisateur. Si ça ne fonctionne pas (accès refusé), expliquez et passez à l'étape 2.

### Etape 2 : Affectez des droits sur la base **GesPerso** à l'utilisateur **User1** :

En session Administrateur, modifiez les propriétés de la connexion **User1** : modifiez le '**mappage de l'utilisateur**' de manière à lui autoriser les droits du rôle '**public**' sur la base de données **GesPerso**.

Basculez en session **User1** et tentez une nouvelle connexion à SQL Server. Tentez d'ouvrir la base **GesPerso** et d'afficher ou de modifier des lignes d'une table (par exemple la table **UNITES**). Expliquez les succès et les échecs.

En session Administrateur, modifiez à nouveau les droits d'accès de l'utilisateur **User1** sur la base **GesPerso** : accordez tout d'abord le rôle **db\_datareader** et testez les mêmes manipulations en session **User1**. Expliquez les succès et les échecs.

### Etape 3 : Refaites une création d'utilisateur en affectant des droits de mise à jour sur la base **GesPerso**

Créez l'utilisateur Windows local '**User2**', qui ne soit pas Administrateur ('utilisateur standard', 'utilisateur avec pouvoirs...').

En session Administrateur, à l'aide de manipulations interactives ou de commandes, sous SQL Server Management Studio, créez une nouvelle **Connexion** à votre serveur SQL Server pour cet utilisateur **User2** en précisant les paramètres suivants :

- Authentification : **Windows**
- Base de données par défaut : **GesPerso**
- Rôle du serveur : **public**
- Mappage de l'utilisateur : rôles **db\_datareader** et **db\_datawriter** sur la base **GesPerso**.
- Autres options : par défaut

Basculez en session **User2** et tentez une nouvelle connexion à SQL Server. Tentez d'ouvrir la base **GesPerso** et d'afficher puis de modifier des lignes d'une table (par exemple la table **UNITES**). Expliquez les réactions du SGBD.

*Ces manipulations ont permis d'accorder des droits plus ou moins limités à des utilisateurs particuliers ; il s'agit maintenant de créer des rôles auxquels on pourra associer différents utilisateurs qui bénéficient des mêmes droits (tout comme on le fait avec des groupes d'utilisateurs Windows).*

## 2.4 ACTIVITE 4 - CREATION DE ROLES

En session Administrateur, à l'aide de manipulations interactives ou de commandes, sous SQL Server Management Studio, créez un nouveau **Rôle GesPersoConsult** sur votre base de données **GesPerso** en précisant les paramètres suivants :

- Propriétaire : **db\_securityadmin**
- Eléments sécurisables : sur la base de données **GesPerso**, autoriser **Connecter**, **Exécuter** et **Sélectionner** (beaucoup de manipulations dans l'interface graphique de Management Studio...)
- Autres options : par défaut

Modifiez les rôles de l'utilisateur **User1** pour ne lui octroyer que les rôles **public** et **GesPersoConsult**. Vérifiez les propriétés de l'utilisateur **User1** pour la base de données **GesPerso**, onglet **Appartenance**.

Basculez en session **User1** et effectuez une nouvelle connexion à SQL Server. Tentez d'ouvrir la base **GesPerso** et d'afficher puis de modifier des lignes d'une table (par exemple la table **UNITES**). Expliquez les réactions du SGBD.

En session Administrateur, à l'aide de manipulations interactives ou de commandes, sous SQL Server Management Studio, créez un nouveau **Rôle GesPersoMAJ** sur votre base de données **GesPerso** en précisant les paramètres suivants :

- Propriétaire : **db\_securityadmin**
- Eléments sécurisables : sur la base de données **GesPerso**, autoriser **Connecter**, **Exécuter**, **Insérer**, **Mettre à jour** et **Sélectionner**
- Autres options : par défaut

Modifiez les rôles de l'utilisateur **User2** pour ne lui octroyer que les rôles **public** et **GesPersoMAJ**. Vérifiez de même les propriétés de l'utilisateur **User2** pour la base de données **GesPerso**, onglet **Appartenance**.

Basculez en session **User2** et effectuez une nouvelle connexion à SQL Server. Tentez d'ouvrir la base **GesPerso** et d'afficher puis de modifier des lignes d'une table (par exemple la table **UNITES**). Expliquez les réactions du SGBD.

Pour bien mettre en évidence cette notion de rôle applicable à différents utilisateurs, vous pouvez encore créer un nouvel utilisateur Windows **User3** et lui affecter le même rôle SQL Server **GesPersoMAJ**.

*Voilà l'essentiel concernant la logique des droits d'accès des utilisateurs. Il serait encore possible de créer un conteneur, ce qui s'appelle un 'schéma' dans SQL Server, de manière à englober tous les objets concernant la base de données GesPerso dans une 'enveloppe logique' à laquelle on pourrait associer des droits. On pourrait de même modifier les droits du schéma par défaut **dbo**. Au final, SQL server propose toute une panoplie d'outils pour gérer les droits d'accès et il est important pour le développeur de bien se faire conseiller dans ce domaine.*

Cas GesPerso SQL Server

## 2.5 ACTIVITE 5 - SAUVEGARDE ET RESTAURATION DE LA BASE DE TEST

Mise en œuvre de la sauvegarde/restauration de la base de données avec l'assistant de sauvegarde/restauration accessible depuis SQL server Management Studio (menu **Tâches/Sauvegarder** et **Tâches/Restaurer**).

Effectuez une sauvegarde complète (toutes tables, données et journaux) de la base de données **GesPerso** ; c'est une bonne pratique d'indiquer la date de sauvegarde dans le nom du fichier (par exemple 'sauveGesPerso20160119.bak').

Effectuez des mises à jour de données sur vos tables ou même une suppression de table.

Effectuez ensuite une restauration complète (données et journaux) de la base **GesPerso** à partir de cette dernière sauvegarde et vérifiez l'état de vos données.

*Ce n'est pas plus compliqué que cela ! Sauvegardes et restaurations sont très faciles à exécuter avec les assistants de SQL Server Management Studio et cela est très utile pour restaurer les jeux d'essai de données lors des tests et de la mise au point des programmes d'application.*

## **Œuvre collective de l'AFPA**

### **Equipe de conception**

Patrice François  
CFPA Saint Brieuc Langueux

Benoit Hézard  
CFPA Nice

## **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »



Cas GesPerso SQL Server

afpa © 2016 – Informatique et télécoms – filière étude - développement