

Secteur Tertiaire Informatique  
Filière « Etude et développement »

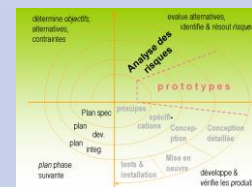
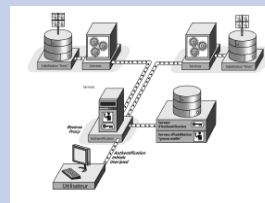
Mettre en place une base de données

**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 - Sauvegarde et Restauration de la base de test ..... | 11 |
| 3. Jeu d'essai.....  | 12 |

## **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 (2016), installée.
- Utilitaire (EDI) SQL Server Management Studio.
- Les scripts LDD SQL du cas GESPERSO générés par votre outil de modélisation Merise.

## 1. RAPPELS

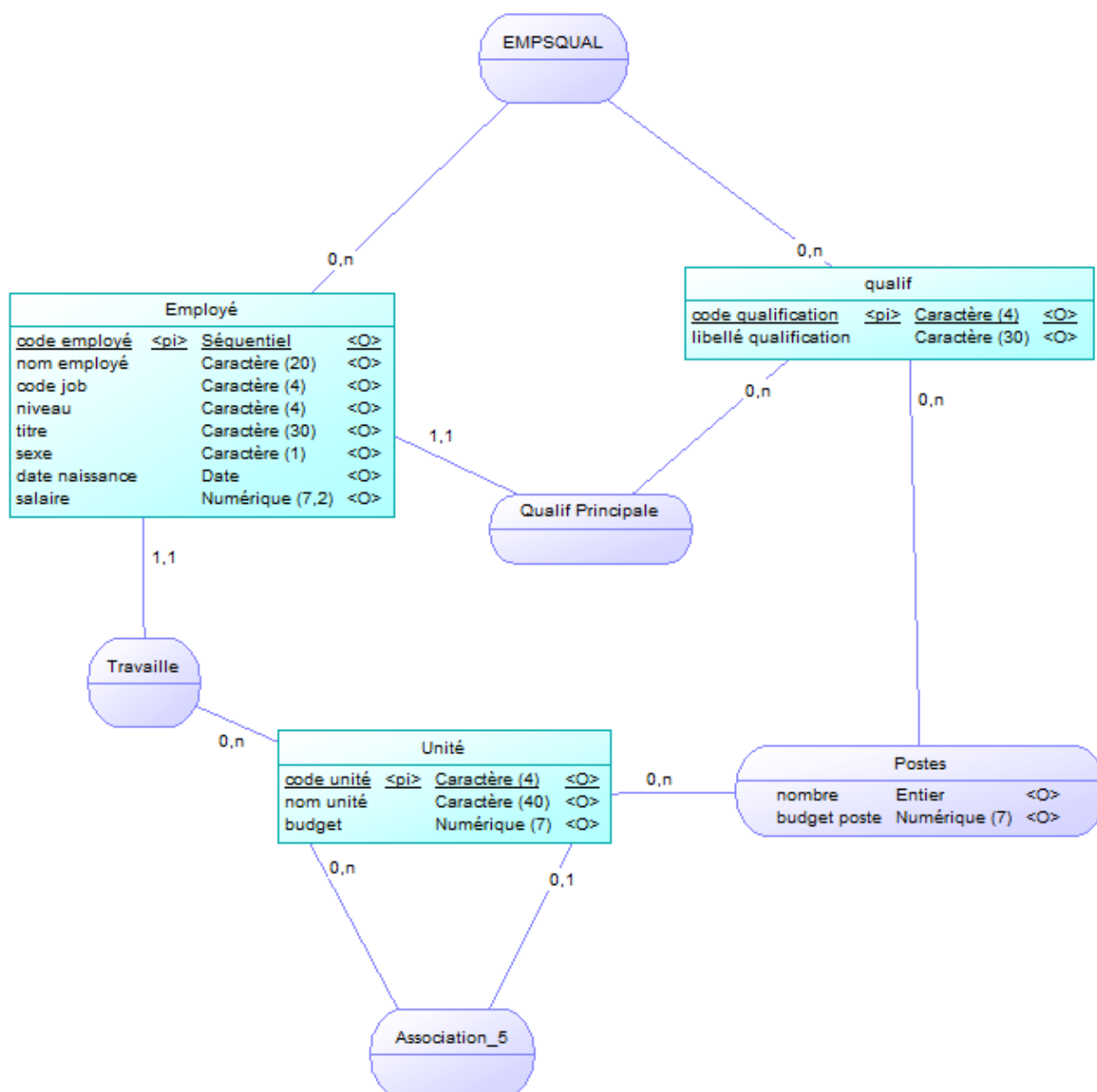
La société DUBILAN et FILS, spécialisée dans la production industrielle, souhaite gérer son personnel à partir des règles de gestion décrites ci-dessous :

- ✓ 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.

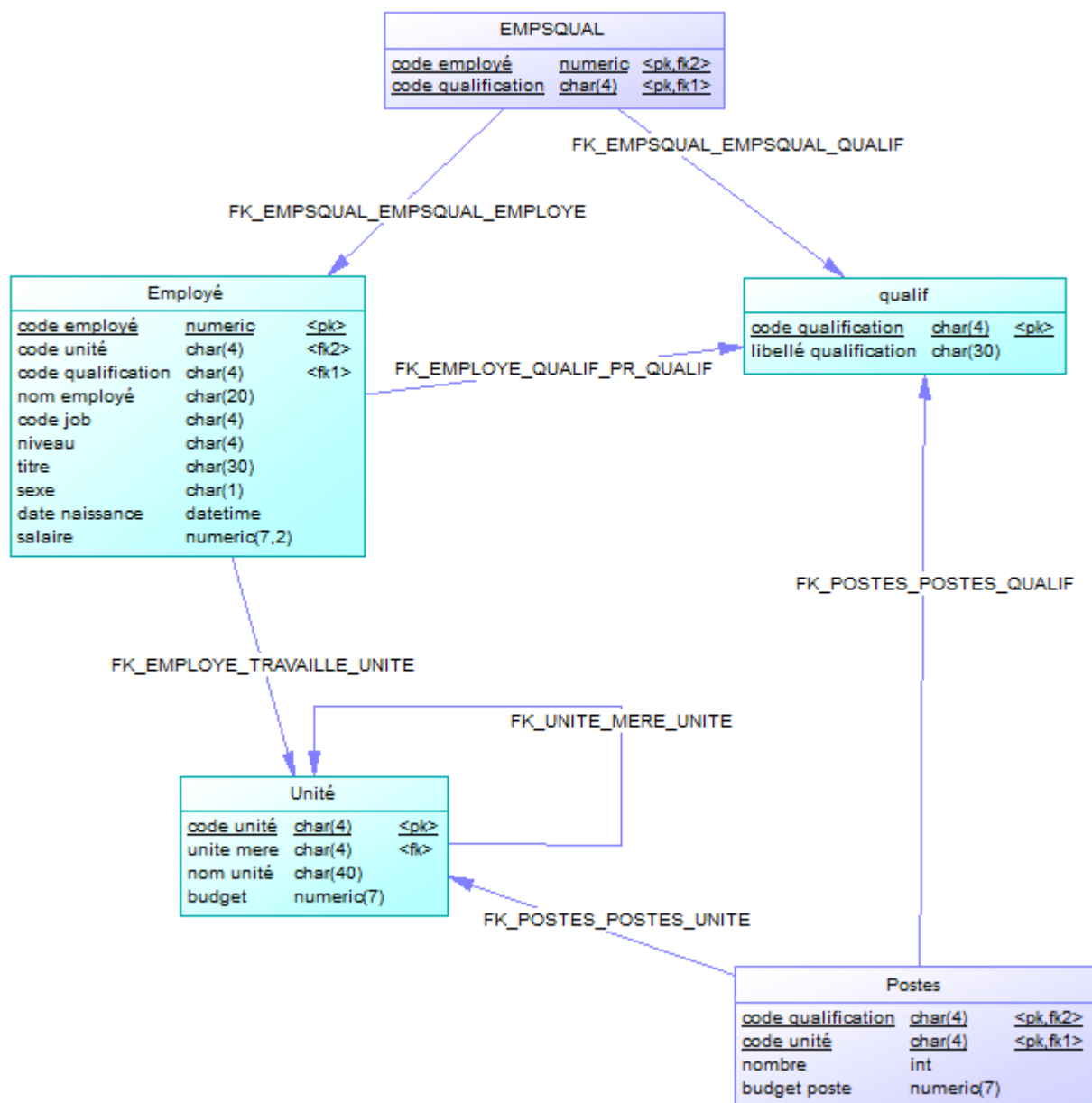
### 1.1 LE DICTIONNAIRE DES DONNEES

| Libellé               | Code        | Type et longueur | Contraintes |
|-----------------------|-------------|------------------|-------------|
| Budget                | BUDGET      | NUM 7,0          |             |
| Budget Poste          | BUDGETPOSTE | NUM 7,0          |             |
| Code Employé          | EMPCOD      | CAR 5            |             |
| Code Job              | JOBCODE     | 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 `EMPLOYEE` 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 - 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.

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 'sauveGesPerso20171024.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.*

-

### 3. JEU D'ESSAI

- Table **QUALIF**

**QUALIF** (**CODQUAL**, LIBQUAL)

| <b>CODQUAL</b> | <b>LIBQUAL</b>         |
|----------------|------------------------|
| 1110           | Secrétaire             |
| 1120           | Ingénieur Mécanicien   |
| 1130           | Ingénieur Electronique |
| 1330           | Dessinateur            |
| 1350           | Responsable            |
| ...            |                        |

- Table **EMPLOYEE**

**EMPLOYEE** (EMPCOD, UNITCOD, CODQUAL, NOM, JOBCODE, NIVEAU, TITRE, SEXE, DATNAISS, SALAIRE)

[illegible]

- Table **POSTES**

**POSTES** (CODQUAL, UNITCOD, NOMBRE, BUDGETPOSTE)

| <u>CODQUAL</u> | <u>UNITCOD</u> | NOMBRE | BUDGETPOSTE (€) |
|----------------|----------------|--------|-----------------|
| 1110           | 2000           | 2      | 1500            |
| 1350           | 2000           | 1      | 50000           |
| 1110           | 2100           | 3      | 1000            |
| 1350           | 2100           | 1      | 2500            |
| 1110           | 2110           | 1      | 1500            |
| 1350           | 2110           | 1      | 3500            |
| 1130           | 2111           | 10     | 5000            |
| 1350           | 2111           | 1      | 4500            |

- Table **EMPSQUAL**

**EMPSQUAL** (EMPCOD, CODQUAL)

| <u>EMPCOD</u> | <u>CODQUAL</u> |
|---------------|----------------|
| 91230         | 1350           |
| 85417         | 1350           |
| ...           |                |

**Œuvre collective de l'AFPA**  
**Equipe de conception**

Patrice François  
 CFPA Saint Briec Langueux

Benoit Hézard  
 Cas GesPerso SQL Server