Module 5

DDL - La gestion des tables

Contenu du module

- Création des tables
- Les types de données SQL Server
- Mise en œuvre de l'intégrité des données
- Modification des tables
- Suppression des tables
- Indexation des données

Création des tables

Instruction du DDL

Nature de l'objet à créer

```
CREATE TABLE table_name
(
        column_name1 data_type(size) [constraint],
        column_name2 data_type(size) [constraint],
        column_name3 data_type(size) [constraint],
        [constraint],
        [constraint],
        ....
);
```

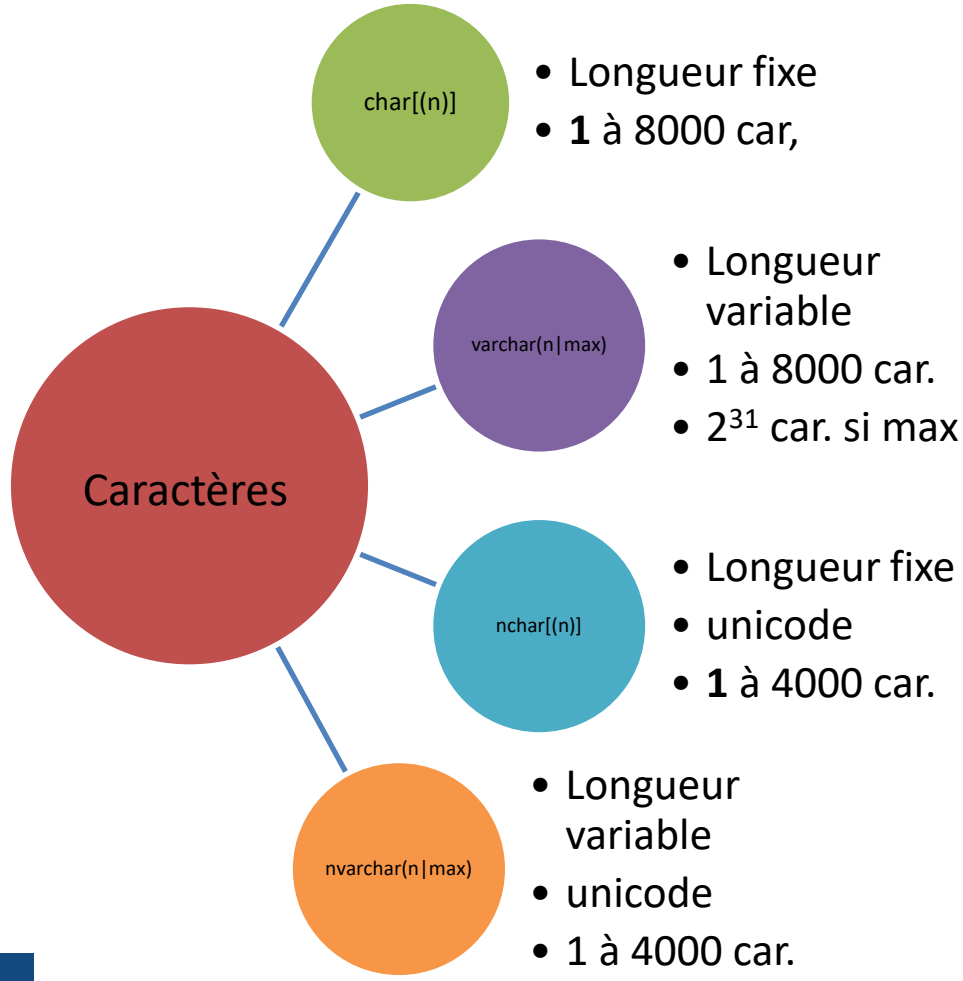
Identifiant :

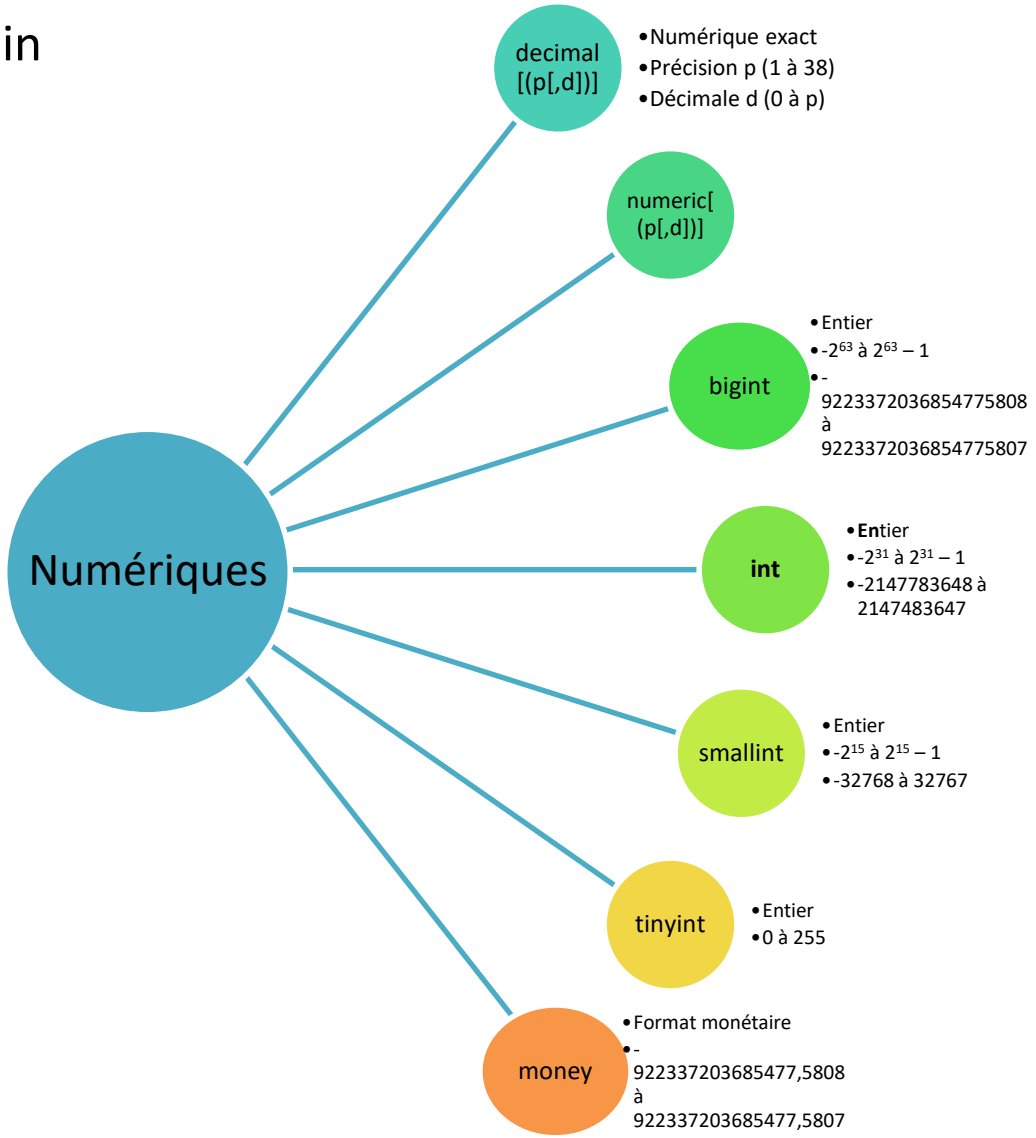
- 1 à 128 caractères
- 1er caractère : lettre, @, _,#
- Puis caractères alphanumériques
- Exemple :

  Gestion_employes,
  [Gestion_employes],
  "Gestion_Employes"

Les types de données SQL Server

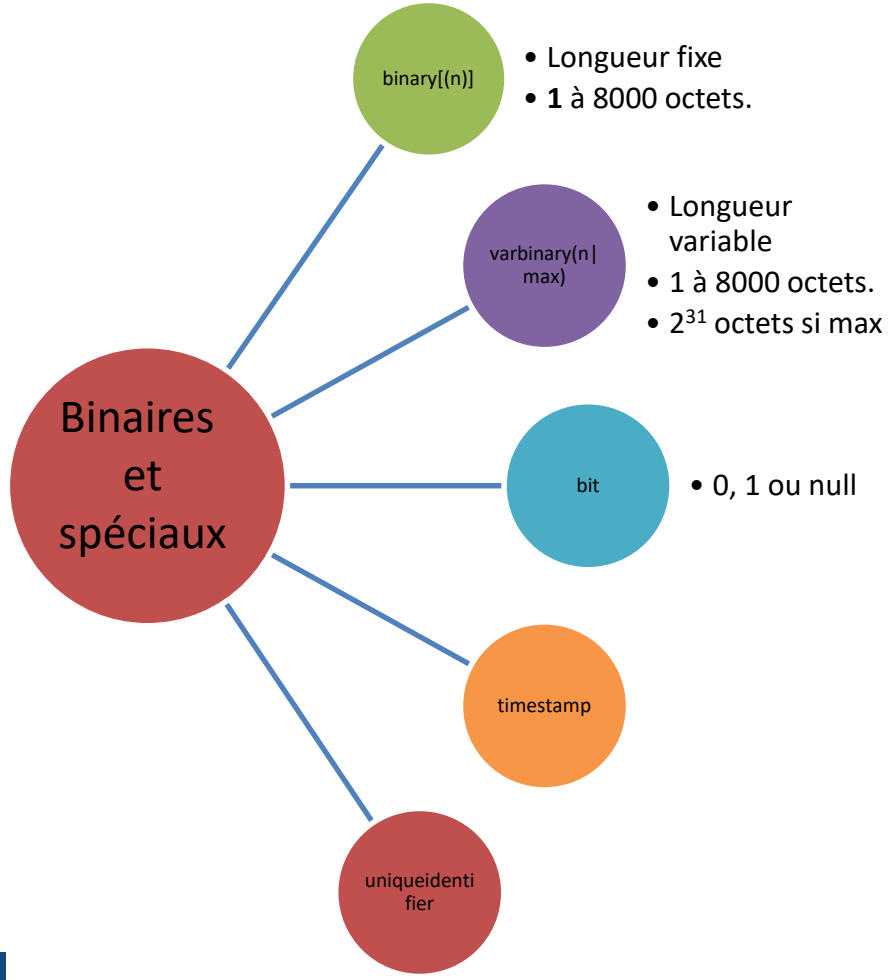
- Choisissez votre type de données en fonction du besoin

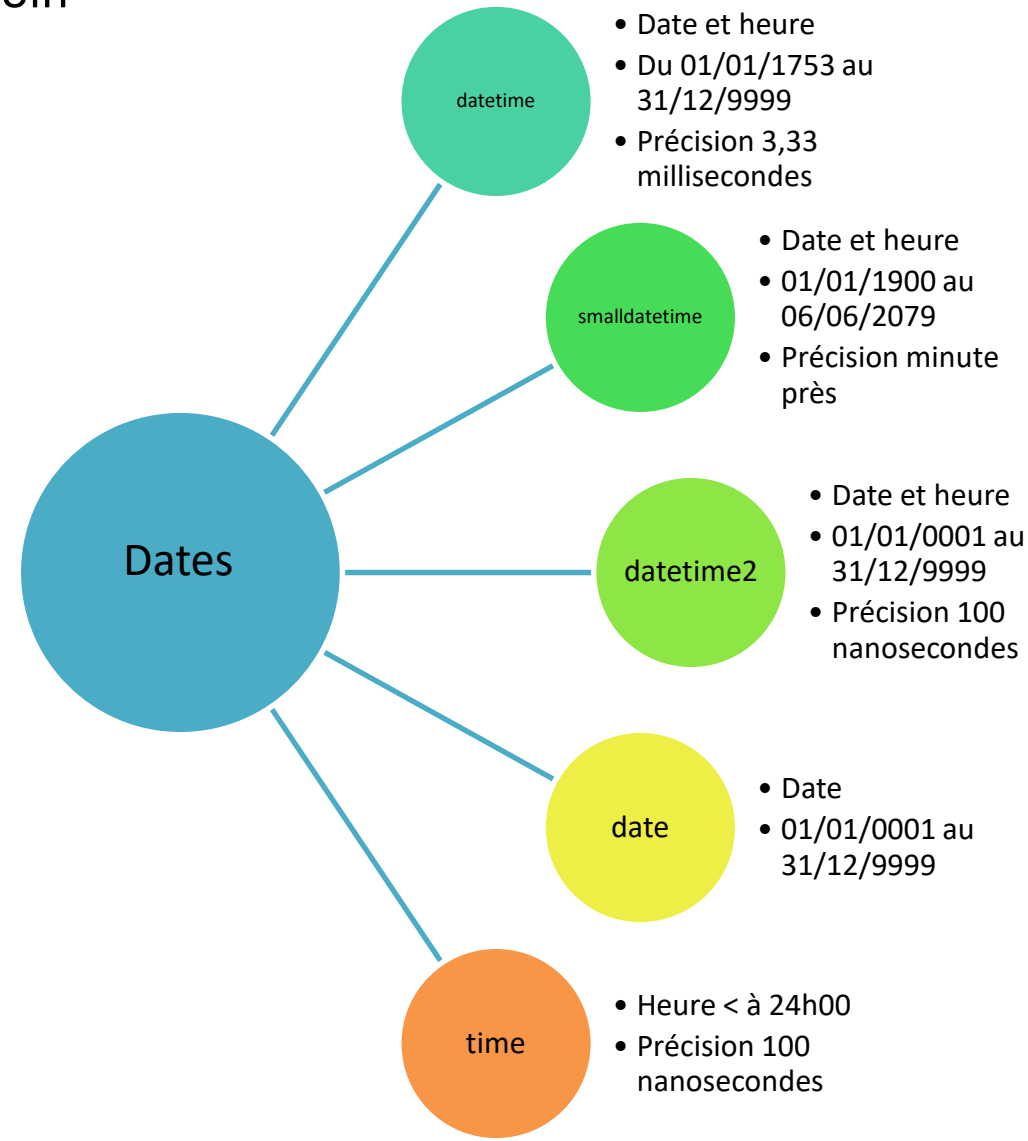




Les types de données SQL Server

- Choisissez votre type de données en fonction du besoin





Compatibilité avec le standard ISO

Synonyme	Type SQL Server
Caractères	
char varying	varchar
character	char(n)
character varying(n)	varchar(n)
national character(n)	nchar(n)
national char(n)	nchar(n)
national character varying(n)	nvarchar(n)
national char varying(n)	nvarchar(n)
national text	Ntext
Numériques	
dec	decimal
double precision	float
integer	int
Binaires	
binary varying	varbinary
Autres	
Rowversion	timestamp

Les types de données

- Les colonnes typées

```
CREATE TABLE Employes(
    CodeEmp INT ,
    Nom VARCHAR(20),
    Prenom CHAR(20),
    DateNaissance DATE,
    DateEmbauche DATE,
    DateModif TIMESTAMP,
    Salaire DECIMAL(8,2),
    CodeService CHAR(5),
    CodeChef INT
);
```

Les types de données SQL Server

- A partir du diagramme proposé dans le fichier en téléchargement *MLD – Gestion Employes.pdf*
  - construisez, sur le modèle de notre table Employes, les tables Services, Conges et Conges_Mens.

Mise en œuvre de l'intégrité des données

- Assurer la cohérence des données
- Appliquer les règles de fonctionnement issues de l'analyse
- Traduction des règles du modèle relationnel

- Réalisable de manière procédurale par les déclencheurs (TRIGGER) ou de manière déclarative, au niveau de la structure de la table elle-même, par les contraintes (CONSTRAINT)

- **Ce cours ne traite que des contraintes.**

```
CREATE TABLE table_name
(
        column_name1 data_type(size) [constraint],
        column_name2 data_type(size) [constraint],
        column_name3 data_type(size) [constraint],
        [constraint],
        [constraint],
        ....
);
```

Contrainte niveau colonne

Contrainte niveau table

Mise en œuvre de l'intégrité des données

- Contrainte de non nullité

```
CREATE TABLE Employes(
    CodeEmp INT             not null,
    Nom VARCHAR(20)         not null,
    Prenom CHAR(20)         null,
    DateNaissance DATE      null,
    DateEmbauche DATE       not null,
    DateModif TIMESTAMP     null,
    Salaire DECIMAL(8,2)    not null,
    Codeservice CHAR(5)     not null,
    CodeChef INT            null
);
```

Mise en œuvre de l'intégrité des données

- Valeur par défaut

```
CREATE TABLE Employes(
    CodeEmp INT             not null,
    Nom VARCHAR(20)         not null,
    Prenom CHAR(20)         null,
    DateNaissance DATE      null,
    DateEmbauche DATE       not null CONSTRAINT DF_Employes_DateEmbauche DEFAULT getdate(),
    DateModif TIMESTAMP     null,
    Salaire DECIMAL(8,2)    not null CONSTRAINT DF_Employes_Salaire DEFAULT 0,
    Codeservice CHAR(5)     not null,
    CodeChef INT            null
);
```

Mise en œuvre de l'intégrité des données

- Clé primaire

```
CREATE TABLE Employes(
    CodeEmp INT             not null CONSTRAINT PK_Employes PRIMARY KEY,
    Nom VARCHAR(20)         not null,
    Prenom CHAR(20)         null,
    DateNaissance DATE      null,
    DateEmbauche DATE       not null CONSTRAINT DF_Employes_DateEmbauche DEFAULT getdate(),
    DateModif TIMESTAMP     null,
    Salaire DECIMAL(8,2)    not null CONSTRAINT DF_Employes_Salaire DEFAULT 0,
    Codeservice CHAR(5)     not null,
    CodeChef INT            null
);
```

ou

```
CREATE TABLE Employes(
    CodeEmp INT             not null,
    Nom VARCHAR(20)         not null,
    Prenom CHAR(20)         null,
    DateNaissance DATE      null,
    DateEmbauche DATE       not null CONSTRAINT DF_Employes_DateEmbauche DEFAULT getdate(),
    DateModif TIMESTAMP     null,
    Salaire DECIMAL(8,2)    not null CONSTRAINT DF_Employes_Salaire DEFAULT 0,
    Codeservice CHAR(5)     not null,
    CodeChef INT            null,
    CONSTRAINT PK_Employes PRIMARY KEY(CodeEmp)
);
```

Une colonne de type compteur

- La propriété IDENTITY
  - Valeurs générées à la création d'une ligne
  - 1 seule colonne IDENTITY par table
  - Peut produire des trous dans la numérotation

```
CodeEmp INT          not null IDENTITY(1,1),
```

Mise en œuvre de l'intégrité des données

- Clés secondaires

```
CREATE TABLE Services(
    CodeService char(5)      not null,
    Libelle varchar(30)      not null CONSTRAINT UN_Services_Libelle UNIQUE,
    CONSTRAINT PK_Services PRIMARY KEY(CodeService)
);
```

ou

```
CREATE TABLE Services(
    CodeService char(5)      not null,
    Libelle varchar(30)      not null,
    CONSTRAINT PK_Services PRIMARY KEY(CodeService),
    CONSTRAINT UN_Services_Libelle UNIQUE(Libelle)
);
```

Mise en œuvre de l'intégrité des données

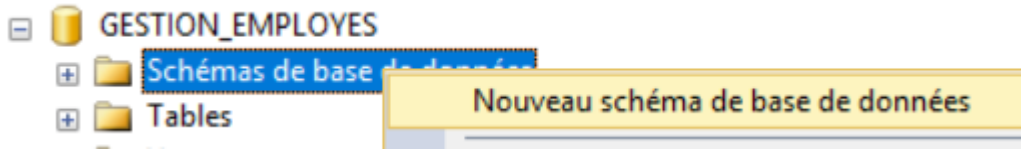
- Contrainte de validation

```
CREATE TABLE Employes(
    CodeEmp INT             not null,
    Nom VARCHAR(20)         not null,
    Prenom CHAR(20)         null,
    DateNaissance DATE      null,
    DateEmbauche DATE       not null CONSTRAINT DF_Employes_DateEmbauche DEFAULT getdate(),
    DateModif TIMESTAMP     null,
    Salaire DECIMAL(8,2)    not null CONSTRAINT DF_Employes_Salaire DEFAULT 0
                                     CONSTRAINT CK_Employes_Salaire CHECK (salaire >= 0),
    Codeservice CHAR(5)     not null,
    CodeChef INT            null,
    CONSTRAINT PK_Employes PRIMARY KEY(CodeEmp),
    CONSTRAINT CK_Employes_VerifDate CHECK (DateNaissance is null or DateEmbauche >= DateNaissance)
);
```

Mise en œuvre de l'intégrité des données

- En vous appuyant sur le diagramme proposé dans le fichier en téléchargement *MLD – Gestion Employes.pdf,* appliquez les contraintes suivantes sur les tables :
  - Contraintes de non nullité
  - Valeurs par défaut :
    - 30 jours pour la colonne NbJoursAcquis
    - 0 jour pour la colonne NbJoursPris
  - Clés primaires
  - Contraintes de validation
    - La colonne Mois n'accepte qu'une valeur comprise entre 1 et 12

Visualisation du schéma de la base de données



Employes

	Nom de la colonne	Type condensé	Nullable	Valeur par défaut	Identité
🔑	CodeEmp	int	Non		☑
	Nom	varchar(20)	Non		☐
	Prenom	char(20)	Oui		☐
	DateNaissance	date	Oui		☐
	DateEmbauche	date	Non	(getdate())	☐
	DateModif	timestamp	Oui		☐
	Salaire	decimal(8, 2)	Non	((0))	☐
	CodeService	char(5)	Non		☐
	CodeChef	int	Oui		☐
					☐

Services

	Nom de la colonne	Type condensé	Nullable	Valeur par défaut	Identité
🔑	CodeService	char(5)	Non		☐
	Libelle	varchar(30)	Non		☐
					☐

Conges

	Nom de la colonne	Type condensé	Nullable	Valeur par défaut	Identité
🔑	CodeEmp	int	Non		☐
🔑	Annee	numeric(4, 0)	Non		☐
	NbJoursAcquis	numeric(2, 0)	Oui	((30))	☐
					☐

Conges_Mens

	Nom de la colonne	Type condensé	Nullable	Valeur par défaut	Identité
🔑	CodeEmp	int	Non		☐
🔑	Annee	numeric(4, 0)	Non		☐
🔑	Mois	numeric(2, 0)	Non		☐
	NbJoursPris	numeric(2, 0)	Oui	((0))	☐
					☐

Modification des tables – les colonnes

- Ajouter une colonne

```
ALTER TABLE table_name

        ADD column_name data_type(size) [NULL | NOT NULL]
;
```

- Modifier une colonne

```
ALTER TABLE table_name

        ALTER COLUMN column_name new_data_type(new_size) [NULL | NOT NULL]
;
```

- Supprimer une colonne

```
ALTER TABLE table_name

        DROP COLUMN column_name
;
```

Modification des tables – les contraintes

- Ajouter une contrainte

```
ALTER TABLE table_name

        [WITH CHECK | WITH NOCHECK] ADD new_constraint_table;
```

- Supprimer une contrainte

```
ALTER TABLE table_name

        DROP CONSTRAINT constraint_name
;
```

- Activer/désactiver une contrainte

```
ALTER TABLE table_name

        {CHECK | NOCHECK} CONSTRAINT {ALL | constraint_name}
;
```

Mise en œuvre de l'intégrité référentielle

- Contrainte d'intégrité référentielle

```
ALTER TABLE Employes WITH CHECK ADD
    CONSTRAINT FK_Employes_CodeService FOREIGN KEY (CodeService)
        REFERENCES Services(CodeService),
    CONSTRAINT FK_Employes_CodeChef FOREIGN KEY (CodeChef)
        REFERENCES Employes(CodeEmp);
```

- ON DELETE | ON UPDATE
  - NO ACTION
  - CASCADE
  - SET NULL
  - SET DEFAULT

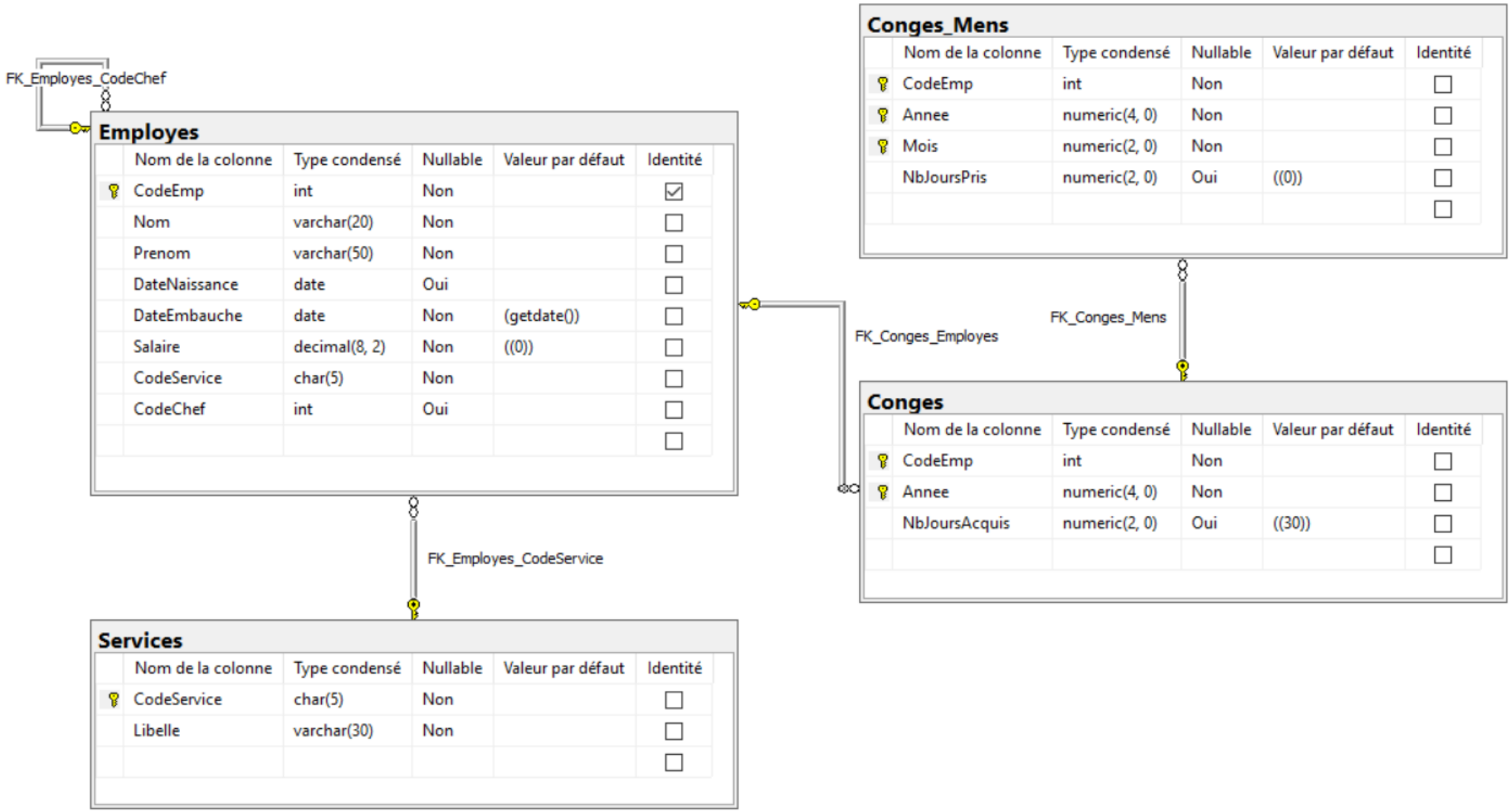
```
ALTER TABLE Conges WITH CHECK ADD
    CONSTRAINT FK_Conges_Employes FOREIGN KEY (CodeEmp)
        REFERENCES Employes(CodeEmp) ON DELETE CASCADE;
```

Faire évoluer la base de données

- En vous appuyant sur le diagramme proposé dans le fichier en téléchargement *MLD – Gestion Employes.pdf,* appliquez les contraintes suivantes sur les tables :
  - Contraintes d'intégrité référentielle entre les tables Conges et Employes, ainsi qu'entre les tables Conges_Mens et Conges
- Préparez un script de suppression de l'ensemble des contraintes d'intégrité référentielle.

Schéma de la base de données

- Etat de la base Gestion_Employes après modifications



Suppression des tables

- Supprimer la structure et les données

```
DROP TABLE table_name [,table_name];
```

Tenir compte de l'intégrité référentielle

Suppression des tables

- Complétez votre script de suppression des contraintes d'intégrité référentielle afin qu'il se termine par la suppression physique des tables.

Indexation des données

- Pourquoi indexer les données ?
  - Améliorer les performances d'accès aux informations en base dans le cadre d'une extraction ou d'une mise à jour.

La gestion des index

- Une bonne stratégie :
  - Il est préférable d'avoir moins d'index que trop d'index,
  - Les index doivent être le plus large possible afin d'être utilisables par plusieurs requêtes,
  - Il faut s'assurer que les requêtes utilisent bien les index.

- Automatiquement créé pour :
  - Une contrainte primary key (index CLUSTERED),
  - Une contrainte unique.

- A définir généralement sur :
  - Une contrainte foreign key,
  - Sur les colonnes de recherche et de tri.

La gestion des index

- Créer un index

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name
ON table_name(column_name [ASC | DESC] [,column_name]);
```

- Supprimer un index

```
DROP INDEX index_name ON table_name;
```

- Reconstruire les index

```
ALTER INDEX { index_name | ALL } ON table_name REBUILD;
```

```
CREATE NONCLUSTERED INDEX FK_Employes_Services
    ON Employes(CodeService ASC);
```