Bill Bouilly

Sébastien Bages

**UTC504**

**Mini-Projet**

**Gestion des factures**

Novembre 2021

Sommaire

[1 - Modèle entité association (MCD)](#_xildqe47to3b) **3**

[1 - 1 - Dictionnaire de données](#_r053grc1fyae) 4

[1 - 2 - Hypothèses](#_tlv7qtg0ymul) 8

[1 - 3 - Contraintes](#_k2oi7ifjurvi) 8

[2 - Modèle logique de données (MLD)](#_k8r8uciyftw9) 8

[2 - 1 - Schéma](#_9xn1q24oxl57) 9

[2 - 2 - MLD textuel](#_p7yrtx5ukckc) 10

[3 - Création BDD](#_3zuqaufgp9sj) **17**

[3 - 1 - Exemple création de tables](#_26hpqltrs6pu) 18

[4 - Insertion de données de tests](#_bi2yykyv3rsj) **19**

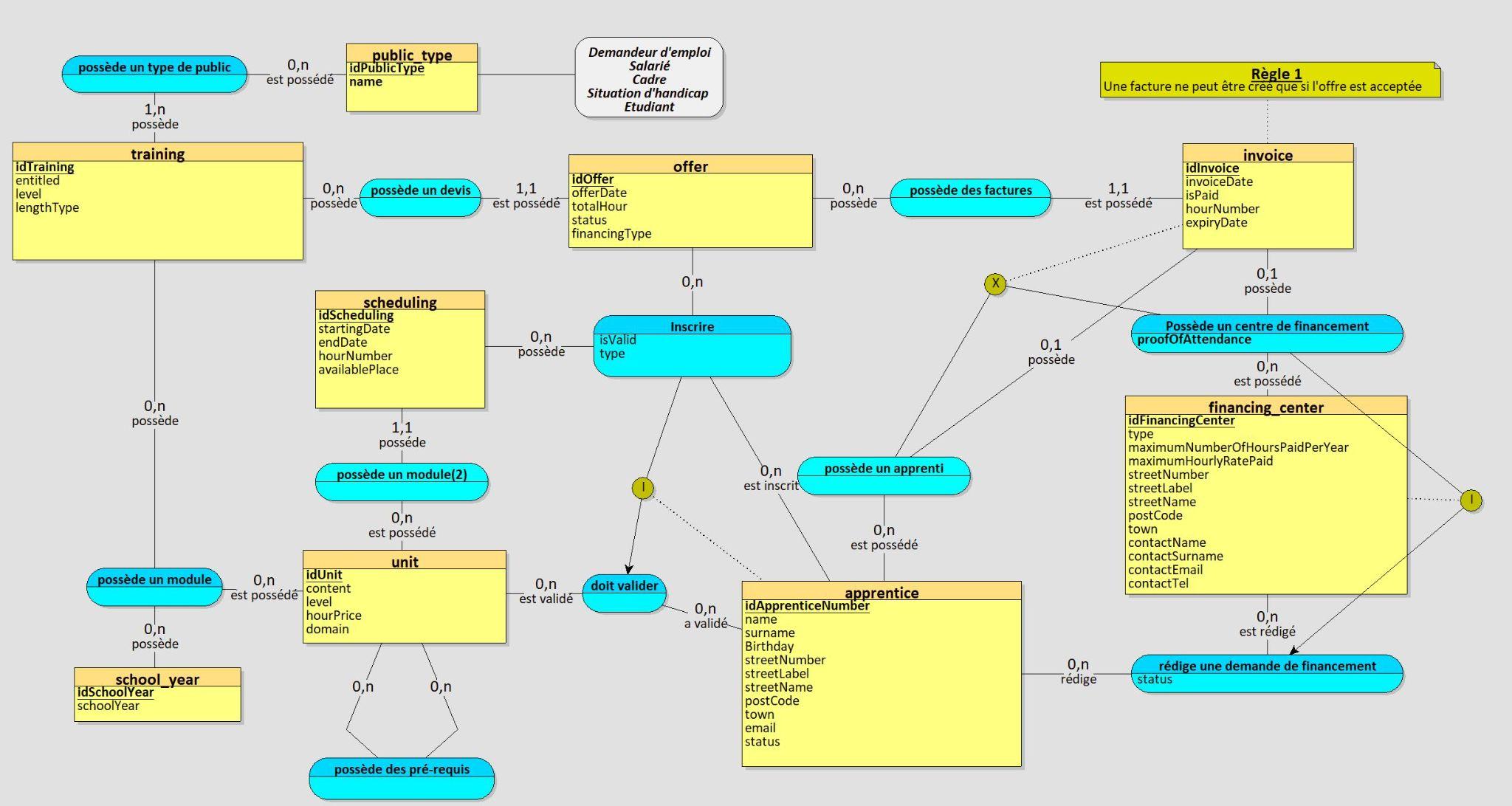
[4 - 1 - Requête](#_1c05udrd8me1) 20

# 

# 1 - Modèle entité association (MCD)

Nous avons utilisé le logiciel de conception [Looping](https://www.looping-mcd.fr/).

Le graphique est disponible [ICI](https://drive.google.com/file/d/1i0jM78_GME9szJzSXYpYxv5EZ1ceSAKt/view?usp=sharing).



## 1 - 1 - Dictionnaire de données

Si besoin, le dictionnaire est disponible [ICI](https://docs.google.com/spreadsheets/d/1-ykBeQWDUmpMZJ9d9xqtEoVCokr2HfQCFFSo8wEIBY8/edit?usp=sharing)

| **Code Table** | **Désignation table** | **Code mnémonique** | **Désignation** | **Type** | **Commentaires** | **Contraintes, règles de calcul** |
| --- | --- | --- | --- | --- | --- | --- |
| **invoice** | **Facture** | **idInvoice** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **invoiceDate** | Date | Date | Date de la facture | NOT NULL |
| **isPaid** | Payé ? | Booléen | Indique si la facture est payée | NOT NULL |
| **hourNumber** | Nombre d'heures | Entier | Nombre d'heures facturé | NOT NULL |
| **expiryDate** | Date | Date | Date d'expiration de la facture | NOT NULL |
| **invoice\_financing\_center** | **Relation**  **facture et centre financement** | **proofOfAttendance** | Preuve de présence | Chaîne | Preuve de présence nécessaire au réglement de la facture | NOT NULL |
| **offer** | **Devis** | **idOffer** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **offerDate** | Date | Date | Date du devis | NOT NULL |
| **totalHour** | Nombre d'heures | Entier | Nombre d'heures proposé | NOT NULL |
| **status** | Statut | Enum | Status : accepté, réfusé ou en cours | NOT NULL |
| **financingType** | Type de financement | Enum | 3 types possibles : personnel, centre de financement, partagé | NOT NULL |
| **financing\_center** | **Centre financier** | **idFinancingCenter** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **type** | Type | Chaîne | Type de centre | / |
| **maximumNumberOfHoursPaidPerYear** | Nombre d'heures | Entier | Nombre d'heures payés par an | NOT NULL |
| **maximumHourlyRatePaid** | Taux horaires | Entier | Taux horaires maximal payé par heure | NOT NULL |
| **adress** | Adresse | Chaîne | Adresse | / |
| **contactName** | Nom | Chaîne | Nom du contact | / |
| **contactSurname** | Prénom | Chaîne | Prénom du contact | / |
| **contactEmail** | Email | Chaîne | Email du contact | / |
| **contactTel** | Téléphone | Chaîne | Téléphone du contact | / |
| **training** | **Formation** | **idTraining** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **entitled** | Intitulé | Chaîne | Indique l'intitulé de la formation | NOT NULL |
| **level** | Niveau | Chaîne | Indique le niveau de la formation | NOT NULL |
| **lengthType** | Durée | Enum | courte, moyenne ou longue | NOT NULL |
| **publicType** | Type de public | Enum | chômeur, employé, cadre, situation de handicap, étudiant | NOT NULL |
| **scheduling** | **Programmation** | **idScheduling** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **startingDate** | Date | Date | Date de début | NOT NULL |
| **endDate** | Date | Date | Date de fin | NOT NULL |
| **hourNumber** | Nombre d'heures | Entier | Nombre total d'heures | NOT NULL |
| **availablePlace** | Place disponibles | Entier | Places disponibles | NOT NULL |
| **apprentice** | **Apprenti** | **idApprenticeNumber** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **name** | Nom | Chaîne | Nom de l'apprenti | NOT NULL |
| **surname** | Prénom | Chaîne | Prénom de l'apprenti | NOT NULL |
| **birthday** | Date de naissance | Date | Date de naissance de l'apprenti | NOT NULL |
| **streetNumber** | N° de voie | Chaîne | Numéro de voie, rue, ... | / |
| **streetLabel** | Libellé de la voie | Chaîne | Libellé de la voie, rue, ... | NOT NULL |
| **streetName** | Nom de la voie | Chaîne | Nom de la voie, rue, ... | NOT NULL |
| **postCode** | Code postal | Chaîne | Code postal de la ville | NOT NULL |
| **town** | Ville | Chaîne | Ville de résidence | NOT NULL |
| **email** | Email | Chaîne | Email de l'apprenti | NOT NULL |
| **status** | Statut | Enum | jeune diplômé, employé, sans emploi | NOT NULL |
| **school\_year** | **Année scolaire** | **idSchoolYear** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **schoolYear** | Couple années | Chaîne | Indique l'année scolaire (aaaa-aaaa) | NOT NULL |
| **unit** | **Module de formation** | **idUnit** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **content** | Contenu | Chaîne | Contenu du module de formation | NOT NULL |
| **level** | Niveau | Entier | Niveau du module | NOT NULL |
| **hourPrice** | Taux horaire | Monétaire | Prix par heure d'un module | NOT NULL |
| **domain** | Domaine | Enum | informatique, économie, histoire | NOT NULL |
| **register** | **inscription** | **isValid** | Validité | Booléen | Validité de l'inscription | NOT NULL |
| **type** | type | Enum | Type d'inscription : formation continue, validation d'acquis, apprentissage ou autres | NOT NULL |
| **request\_funding** | **Demande de financement** | **status** | Statut | Enum | 3 statuts possibles : rejeté, accepté partiellement, accepté complètement | NOT NULL |
| **public\_type** | **Type de public** | **idPublicType** | Identifiant | Entier | Clé primaire | AUTO INCREMENT, NOT NULL, UNIQUE |
| **name** | Nom du type | Chaîne | Nom du type unique | NOT NULL |

## 1 - 2 - Hypothèses

* Nous pouvons définir des prix de l’heure différents selon les modules.
* Le domaine sera un attribut de la table module.

## 1 - 3 - Contraintes

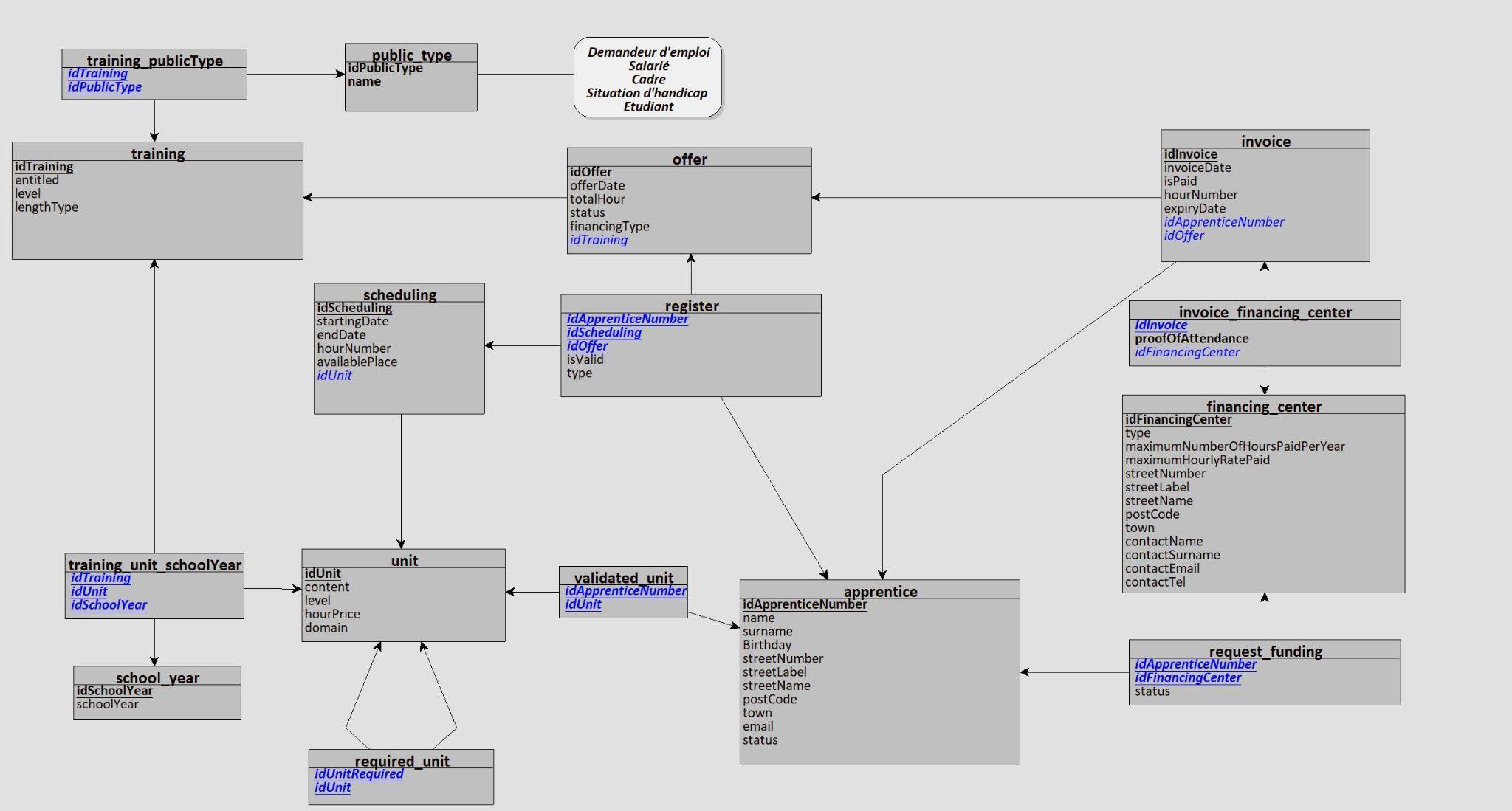
## 

* Pour qu'une facture possède un centre de financement, une demande de financement associé doit être validée.
* Pour s'inscrire un à un module, l'apprenti doit avoir validé les modules pré-requis.
* Une facture est adressée soit à un apprenti ou à un centre de financement.
* Le nombre d'heures facturées est égal aux nombre d'heures total des programmations où l'apprenti est inscrit.
* Le nombre de places disponibles doit être mis à jour au fur et à mesure des inscriptions.
* Pour générer une facture il faut que le devis possède un statut “accepté”.
* Toutes les clés primaires des tables sont auto-incrémentées et non nulles.

## 2 - Modèle logique de données (MLD)

## 2 - 1 - Schéma

Le schéma est disponible [ICI](https://drive.google.com/file/d/1CGnyVTG8aJzCnxcaAK1jjDYndIc1n7vJ/view?usp=sharing)



## 2 - 2 - MLD textuel

Le MLD est disponible [ICI](https://drive.google.com/file/d/1Nm9PloroOynJFkpBAzKE_TCgMsH0S6P_/view?usp=sharing)

**apprentice** = (**idApprenticeNumber** INT AUTO\_INCREMENT, name VARCHAR(20) , surname VARCHAR(20) , Birthday DATE, streetNumber VARCHAR(50) , streetLabel VARCHAR(50) , streetName VARCHAR(50) , postCode VARCHAR(5) , town VARCHAR(50) , email VARCHAR(50) , status ENUM('youngGraduate','employee','jobless'));

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**training** = (**idTraining** INT AUTO\_INCREMENT, entitled VARCHAR(100) , level VARCHAR(50) , lengthType ENUM('short','medium','long'));

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**unit** = (**idUnit** INT AUTO\_INCREMENT, content VARCHAR(100) , level TINYINT, hourPrice DECIMAL(19,4), domain ENUM('computing','economy','history'));

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**school\_year** = (**idSchoolYear** INT AUTO\_INCREMENT, schoolYear VARCHAR(50) );

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

la propriété dépend de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

la propriété non identifiante dépend directement de l’identifiant et non d’une propriété de l’identifiant

**scheduling** = (**idScheduling** INT AUTO\_INCREMENT, startingDate DATE, endDate DATE, hourNumber SMALLINT, availablePlace SMALLINT, #idUnit);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**offer** = (**idOffer** INT AUTO\_INCREMENT, offerDate DATE, totalHour TINYINT, status ENUM('accepted', 'aborted','inProgress'), financingType ENUM('personnel','financingCentre','shared'), #idTraining);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**invoice** = (**idInvoice** INT AUTO\_INCREMENT, invoiceDate DATE, isPaid BOOLEAN, hourNumber TINYINT, expiryDate DATE, #idApprenticeNumber\*, #idOffer);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**financing\_center** = (**idFinancingCenter** INT AUTO\_INCREMENT, type VARCHAR(50) , maximumNumberOfHoursPaidPerYear INT, maximumHourlyRatePaid INT, streetNumber VARCHAR(50) , streetLabel VARCHAR(50) , streetName VARCHAR(50) , postCode VARCHAR(50) , town VARCHAR(50) , contactName VARCHAR(50) , contactSurname VARCHAR(50) , contactEmail VARCHAR(50) , contactTel VARCHAR(10) );

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**public\_type = (idPublicType** INT, name VARCHAR(50) );

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

la propriété dépend de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

la propriété non identifiante dépend directement de l’identifiant et non d’une propriété de l’identifiant

**training\_unit\_schoolYear** = (**#idTraining, #idUnit, #idSchoolYear**);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

il n’est composé que de la clé primaire

est en 3ème forme normale

il est en deuxième forme normale

il n’est composé que de la clé primaire

**register** = (**#idApprenticeNumber, #idScheduling, #idOffer**, isValid BOOLEAN, type ENUM('continuousTraining','priorLearningAssessment','learning','other'));

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**required\_unit** = (**#idUnitRequired, #idUnit**);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

il n’est composé que de la clé primaire

est en 3ème forme normale

il est en deuxième forme normale

il n’est composé que de la clé primaire

**validated\_unit** = (**#idApprenticeNumber, #idUnit**);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

il n’est composé que de la clé primaire

est en 3ème forme normale

il est en deuxième forme normale

il n’est composé que de la clé primaire

**request\_funding** = (**#idApprenticeNumber, #idFinancingCenter,** status ENUM('rejected','partiallyGranted','fullyGranted'));

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

la propriété dépend de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

la propriété non identifiante dépend directement de l’identifiant et non d’une propriété de l’identifiant

**invoice\_financing\_center** = (**#idInvoice**, proofOfAttendance VARCHAR(50) , #idFinancingCenter);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

les propriétés dépendent de la totalité de l’identifiant

est en 3ème forme normale

il est en deuxième forme normale

les propriétés non identifiantes dépendent directement de l’identifiant et non d’une propriété de l’identifiant

**training\_publicType** = **(#idTraining\_possède**, **#idPublicType\_\_est\_possédé**);

est en première forme normale :

Il ne contient pas de propriété répétitive ou décomposable.

est en deuxième forme normale :

il est en première forme normale

il n’est composé que de la clé primaire

est en 3ème forme normale

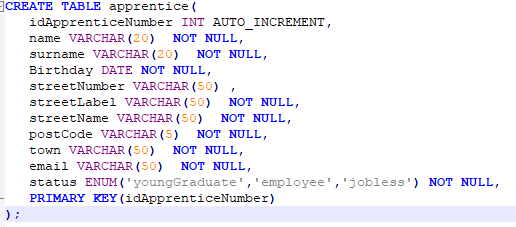
il est en deuxième forme normale

il n’est composé que de la clé primaire

# 3 - Création BDD

Nous avons utilisé le SGBDR “[MySql](https://www.mysql.com/fr/)” et avons exécuté le script SQL généré par notre logiciel de conception.

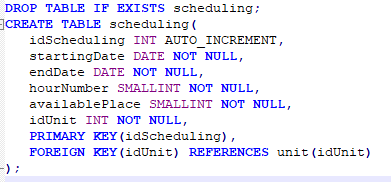
## 3 - 1 - Exemple création de tables



Contraintes misent en place:

- idApprenticeNumber : identifiant de type entier avec contrainte AUTO\_INCREMENT et PRIMARY KEY.

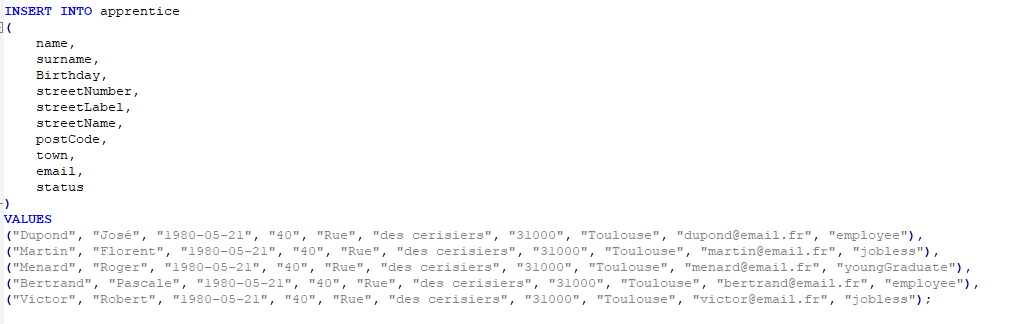
- Les attributs ont une contrainte NOT NULL.



- idUnit a les contraintes de clé étrangère FOREIGN KEY et REFERENCES.

Le fichier de création des tables est disponible [ICI](https://drive.google.com/file/d/1pqXJwwYSRxH3-jif9P3pZbcGBPwwmN18/view?usp=sharing).

# 4 - Insertion de données de tests



Le fichier pour la création de l’ensemble des données est disponible [ICI](https://drive.google.com/file/d/1rGP2fo6gAgwwThekNgMd2YUs5xC_QJfj/view?usp=sharing).

## 4 - 1 - Requête

Voici une requêtes permettant d’obtenir la liste des modules pour lesquelles est inscrit l’apprenti N°3 (Martin) :

SELECT

apprentice.name AS nom\_apprenti,

apprentice.surname AS prenom\_apprenti,

scheduling.startingDate AS date\_debut,

scheduling.endDate AS date\_fin,

scheduling.hourNumber AS nombre\_heures,

unit.hourPrice AS prix\_heure,

unit.domain AS domaine,

unit.content AS contenu

FROM apprentice

INNER JOIN register ON apprentice.idApprenticeNumber = register.idApprenticeNumber

INNER JOIN scheduling ON register.idScheduling = scheduling.idScheduling

INNER JOIN unit ON unit.idUnit = scheduling.idUnit

WHERE apprentice.name = 'Martin'

Nous obtenons le résultat suivant :

