

Bill Bouilly  
Sébastien Bages

**UTC504**  
**Mini-Projet**  
**Gestion des factures**

Novembre 2021

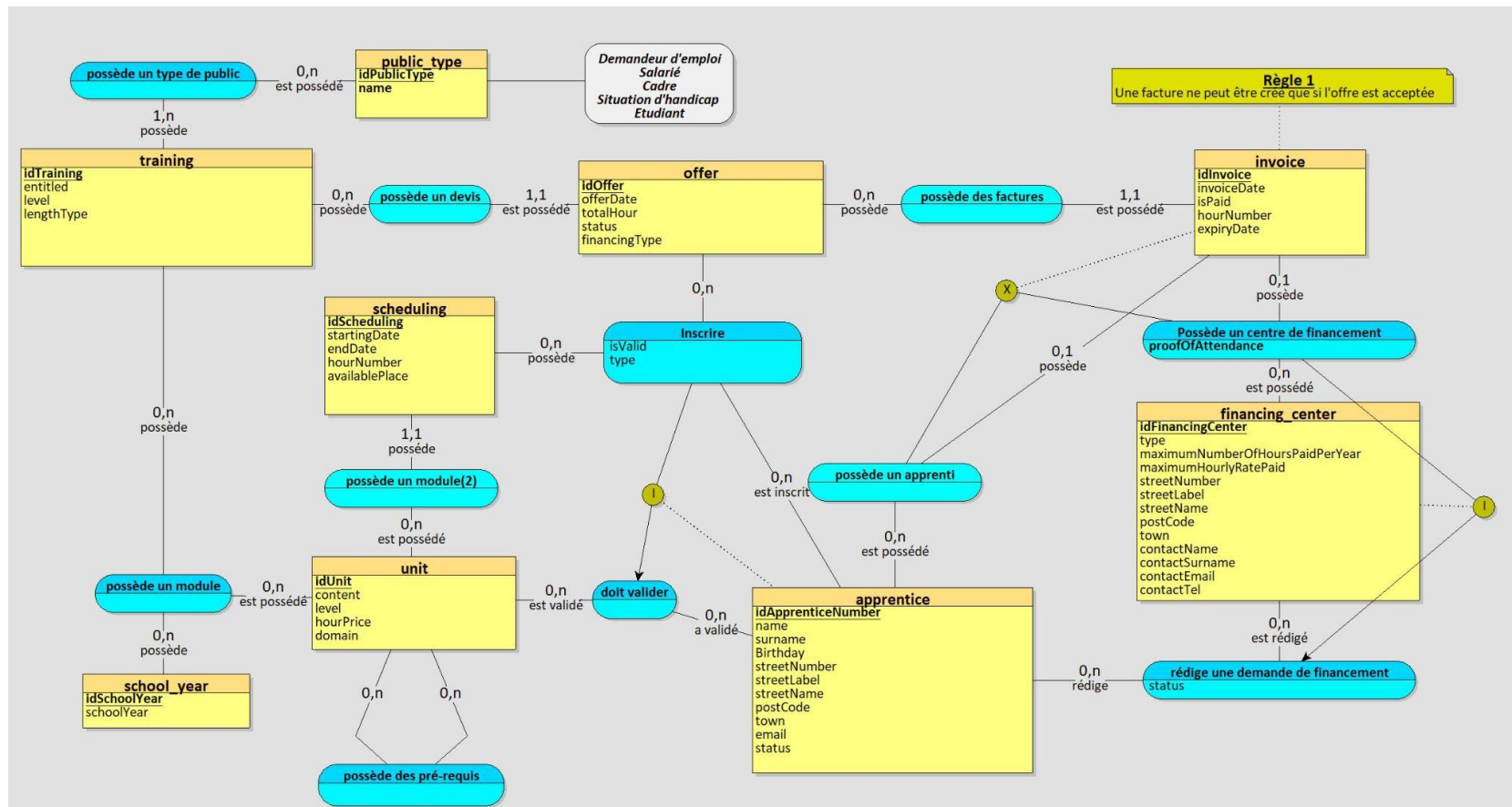
# Sommaire

1 - Modèle entité association (MCD)	<b>3</b>
1 - 1 - Dictionnaire de données	4
1 - 2 - Hypothèses	7
1 - 3 - Contraintes	7
2 - Modèle logique de données (MLD)	7
2 - 1 - Schéma	8
2 - 2 - MLD textuel	9
3 - Création BDD	<b>16</b>
3 - 1 - Exemple création de tables	17
4 - Insertion de données de tests	<b>18</b>
4 - 1 - Requête	19

# 1 - Modèle entité association (MCD)

Nous avons utilisé le logiciel de conception [Looping](#).

Le graphique est disponible [ICI](#).



## 1 - 1 - Dictionnaire de données

Si besoin, le dictionnaire est disponible [ICI](#)

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 règlement 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, refusé ou en cours	NOT NULL
		financingType	Type de financement	Enum	3 types possibles : personnel, centre de financement, partagé	NOT NULL
financing_center	Centre financé	idFinancingCenter	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		type	Type	Chaîne	Type de centre	/

		<b>maximumNumberOfHoursPaidPerYear</b>	Nombre d'heures	Entier	Nombre d'heures payés par an	NOT NULL
		<b>maximumHourlyRatePaid</b>	Taux horaires	Entier	Taux horaires maximal payé par heure	NOT NULL
		<b>adress</b>	Adresse	Chaîne	Adresse	/
		<b>contactName</b>	Nom	Chaîne	Nom du contact	/
		<b>contactSurname</b>	Prénom	Chaîne	Prénom du contact	/
		<b>contactEmail</b>	Email	Chaîne	Email du contact	/
		<b>contactTel</b>	Téléphone	Chaîne	Téléphone du contact	/
<b>training</b>	<b>Formation</b>	<b>idTraining</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>entitled</b>	Intitulé	Chaîne	Indique l'intitulé de la formation	NOT NULL
		<b>level</b>	Niveau	Chaîne	Indique le niveau de la formation	NOT NULL
		<b>lengthType</b>	Durée	Enum	courte, moyenne ou longue	NOT NULL
		<b>publicType</b>	Type de public	Enum	chômeur, employé, cadre, situation de handicap, étudiant	NOT NULL
<b>scheduling</b>	<b>Programmation</b>	<b>idScheduling</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>startingDate</b>	Date	Date	Date de début	NOT NULL
		<b>endDate</b>	Date	Date	Date de fin	NOT NULL
		<b>hourNumber</b>	Nombre d'heures	Entier	Nombre total d'heures	NOT NULL
		<b>availablePlace</b>	Place disponibles	Entier	Places disponibles	NOT NULL
<b>apprentice</b>	<b>Apprenti</b>	<b>idApprenticeNumber</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>name</b>	Nom	Chaîne	Nom de l'apprenti	NOT NULL
		<b>surname</b>	Prénom	Chaîne	Prénom de l'apprenti	NOT NULL
		<b>birthday</b>	Date de naissance	Date	Date de naissance de l'apprenti	NOT NULL

		<b>streetNumber</b>	N° de voie	Chaîne	Numéro de voie, rue, ...	/
		<b>streetLabel</b>	Libellé de la voie	Chaîne	Libellé de la voie, rue, ...	NOT NULL
		<b>streetName</b>	Nom de la voie	Chaîne	Nom de la voie, rue, ...	NOT NULL
		<b>postCode</b>	Code postal	Chaîne	Code postal de la ville	NOT NULL
		<b>town</b>	Ville	Chaîne	Ville de résidence	NOT NULL
		<b>email</b>	Email	Chaîne	Email de l'apprenti	NOT NULL
		<b>status</b>	Statut	Enum	jeune diplômé, employé, sans emploi	NOT NULL
<b>school_year</b>	<b>Année scolaire</b>	<b>idSchoolYear</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>schoolYear</b>	Couple années	Chaîne	Indique l'année scolaire (aaaa-aaaa)	NOT NULL
<b>unit</b>	<b>Module de formation</b>	<b>idUnit</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>content</b>	Contenu	Chaîne	Contenu du module de formation	NOT NULL
		<b>level</b>	Niveau	Entier	Niveau du module	NOT NULL
		<b>hourPrice</b>	Taux horaire	Monétaire	Prix par heure d'un module	NOT NULL
		<b>domain</b>	Domaine	Enum	informatique, économie, histoire	NOT NULL
<b>register</b>	<b>inscription</b>	<b>isValid</b>	Validité	Booléen	Validité de l'inscription	NOT NULL
		<b>type</b>	type	Enum	Type d'inscription : formation continue, validation d'acquis, apprentissage ou autres	NOT NULL
<b>request_funding</b>	<b>Demande de financement</b>	<b>status</b>	Statut	Enum	3 statuts possibles : rejeté, accepté partiellement, accepté complètement	NOT NULL
<b>public_type</b>	<b>Type de public</b>	<b>idPublicType</b>	Identifiant	Entier	Clé primaire	AUTO INCREMENT, NOT NULL, UNIQUE
		<b>name</b>	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.

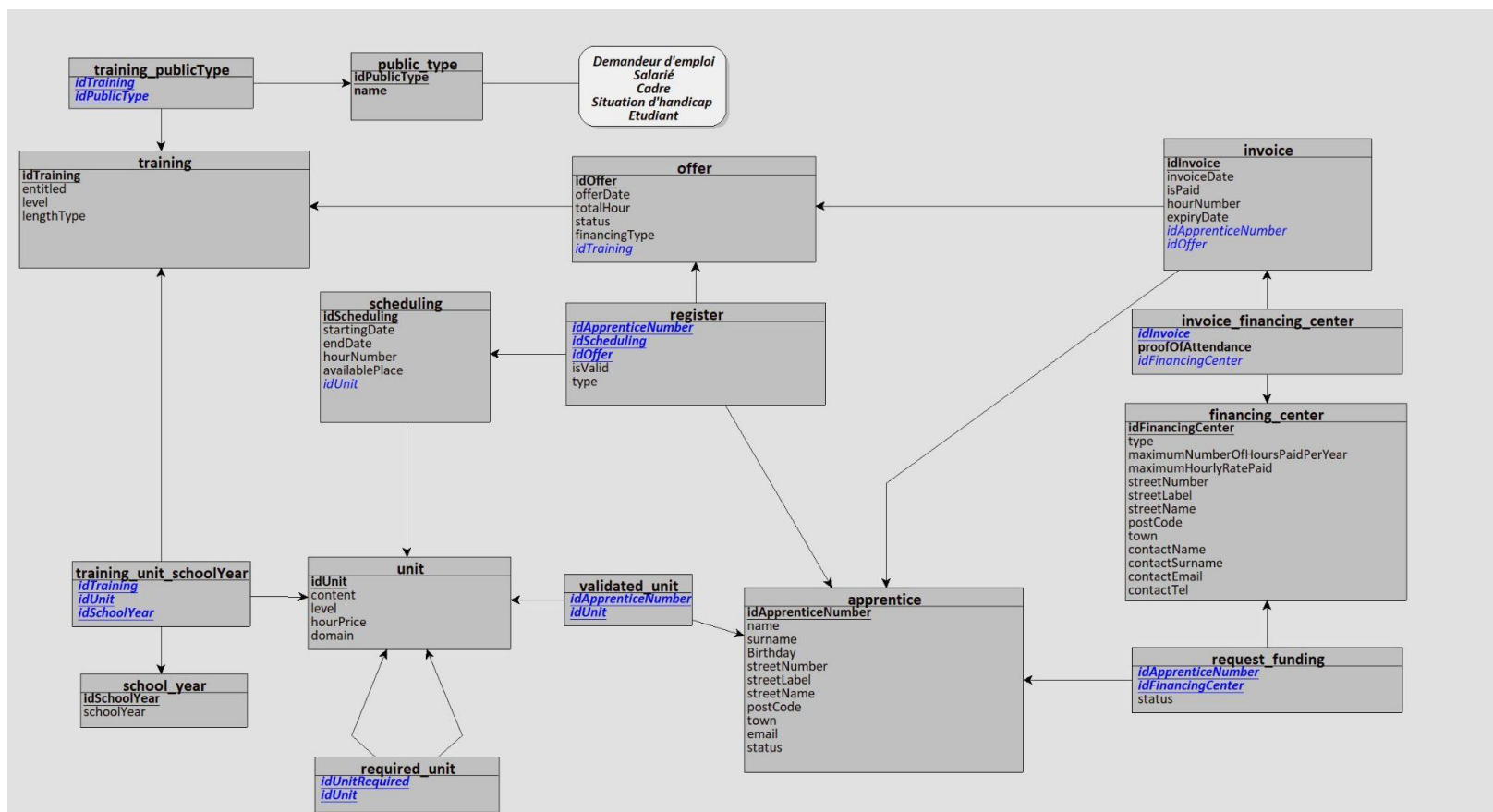
## 1 - 3 - Contraintes

- 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](#)





## 2 - 2 - MLD textuel

Le MLD est disponible [ICI](#)

**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](#)" et avons exécuté le script SQL généré par notre logiciel de conception.



### 3 - 1 - Exemple création de tables

```
CREATE TABLE apprentice(  
    idApprenticeNumber INT AUTO_INCREMENT,  
    name VARCHAR(20) NOT NULL,  
    surname VARCHAR(20) NOT NULL,  
    Birthday DATE NOT NULL,  
    streetNumber VARCHAR(50) ,  
    streetLabel VARCHAR(50) NOT NULL,  
    streetName VARCHAR(50) NOT NULL,  
    postCode VARCHAR(5) NOT NULL,  
    town VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    status ENUM('youngGraduate','employee','jobless') NOT NULL,  
    PRIMARY KEY(idApprenticeNumber)  
);
```

Contraintes misent en place:

- idApprenticeNumber : identifiant de type entier avec contrainte AUTO\_INCREMENT et PRIMARY KEY.
- Les attributs ont une contrainte NOT NULL.

```
DROP TABLE IF EXISTS scheduling;  
CREATE TABLE scheduling(  
    idScheduling INT AUTO_INCREMENT,  
    startingDate DATE NOT NULL,  
    endDate DATE NOT NULL,  
    hourNumber SMALLINT NOT NULL,  
    availablePlace SMALLINT NOT NULL,  
    idUnit INT NOT NULL,  
    PRIMARY KEY(idScheduling),  
    FOREIGN KEY(idUnit) REFERENCES unit(idUnit)  
);
```

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

Le fichier de création des tables est disponible [ICI](#).

## 4 - Insertion de données de tests

```
INSERT INTO apprentice
[ (
    name,
    surname,
    Birthday,
    streetNumber,
    streetLabel,
    streetName,
    postCode,
    town,
    email,
    status
-)
VALUES
("Dupond", "José", "1980-05-21", "40", "Rue", "des cerisiers", "31000", "Toulouse", "dupond@email.fr", "employee"),
("Martin", "Florent", "1980-05-21", "40", "Rue", "des cerisiers", "31000", "Toulouse", "martin@email.fr", "jobless"),
("Menard", "Roger", "1980-05-21", "40", "Rue", "des cerisiers", "31000", "Toulouse", "menard@email.fr", "youngGraduate"),
("Bertrand", "Pascale", "1980-05-21", "40", "Rue", "des cerisiers", "31000", "Toulouse", "bertrand@email.fr", "employee"),
("Victor", "Robert", "1980-05-21", "40", "Rue", "des cerisiers", "31000", "Toulouse", "victor@email.fr", "jobless");
```

Le fichier pour la création de l'ensemble des données est disponible [ICI](#).

## 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 :

nom_apprenti	prenom_apprenti	date_debut	date_fin	nombre_heures	prix_heure	domaine	contenu
Martin	Florent	2021-06-01	2021-06-30	50	53.0000	computing	Module avancé POO
Martin	Florent	2021-06-01	2021-06-30	35	46.0000	computing	Module avancé systèmes
Martin	Florent	2021-06-01	2021-06-30	40	58.0000	computing	Module avancé Réseaux