

3.12

1.

```
CREATE TABLE Professor(ssn CHAR(9),  
    PRIMARY KEY (ssn));  
  
CREATE TABLE Course(courseId INTEGER,  
    PRIMARY KEY (courseId));  
  
CREATE TABLE Semester(semesterId INTEGER,  
    PRIMARY KEY (semesterId));  
  
CREATE TABLE Teaches(ssn CHAR(9),  
    courseId INTEGER,  
    semesterId INTEGER,  
    PRIMARY KEY (ssn,courseId,semesterId),  
    FOREIGN KEY (ssn) REFERENCES Professor,  
    FOREIGN KEY (courseId) REFERENCES Course,  
    FOREIGN KEY (semesterId) REFERENCES Semester);
```

2.

```
CREATE TABLE Teaches(ssn CHAR(9),  
    courseId INTEGER,  
    semester CHAR(10),  
    PRIMARY KEY (ssn,courseId),  
    FOREIGN KEY (ssn) REFERENCES Professor,  
    FOREIGN KEY (courseId) REFERENCES Courses);
```

3. Les tables créées précédemment sont les meilleures que l'on peut produire sans utiliser de contraintes CHECK ou des assertions SQL.

La contrainte de participation totale seulement(1,n) ne peut pas être définie en utilisant seulement les clés primaires and étrangères; on ne peut pas s'assurer que chaque entrée Professeur ait une entrée Teaches.

```
4. CREATE TABLE Professor_teaches(ssn CHAR(9),  
    courseId INTEGER NOT NULL,  
    semester CHAR(10),  
    PRIMARY KEY (ssn),  
    FOREIGN KEY (courseId) REFERENCES Courses);
```

On a plus besoin de la table pour professeur puisque chaque professeur enseigne maximum un cours, on joint la relation teaches et l'entité professeur en une table. Ne pas oublier le NOT NULL pour courseId qui indique qu'on a également une participation totale.

```
5. CREATE TABLE Professor_teaches(ssn CHAR(9),  
    courseId INTEGER NOT NULL,  
    semester CHAR(10),  
    PRIMARY KEY (ssn),  
    FOREIGN KEY (courseId) REFERENCES Courses);
```

On doit enlever cours ici, puisque tous les cours doivent être enseignés et le seul attribut de cours est sa clé. Si on avait d'autres attributs pour cours, on devrait avoir une entité séparée. Dans ce cas, on ne pourrait pas renforcer la contrainte comme quoi chaque cours doit être enseigné par un professeur(sans inclure tout les attributs du cours dans l'entité professeur_teaches ce qui causerait de la redondance si on plusieurs professeurs qui enseigne le même cours).

Note : Si on a deux entités reliées toutes les deux par des relations (1,1), on peut représenter les entités et la relation comme une seule table.

3.16

```
CREATE TABLE Employees (ssn CHAR(9),  
    union_nb INTEGER,  
    PRIMARY KEY (ssn));
```

```
CREATE TABLE Technicien_emp (ssn CHAR(9),  
    Nom CHAR(20),  
    Address CHAR(20),  
    Phone CHAR(10),  
    PRIMARY KEY (ssn),  
    FOREIGN KEY (ssn) REFERENCES Employees,  
    ON DELETE CASCADE);
```

Cascade indique que si une entrée d'employée est supprimée, et qu'une entrée de Technicien lui fait référence, elle est elle aussi supprimée.

```
CREATE TABLE Traffic_control_emp (ssn CHAR(9),  
    Exam_date DATE,  
    PRIMARY KEY (ssn),  
    FOREIGN KEY (ssn) REFERENCES Employees,  
    ON DELETE CASCADE);
```

```
CREATE TABLE Model (model_no INTEGER,  
    Capacity INTEGER,  
    Weight INTEGER,  
    PRIMARY KEY (model_no));
```

```
CREATE TABLE expert(ssn CHAR(9),
    Model_no INTEGER,
    PRIMARY KEY (ssn,model_no),
    FOREIGN KEY (ssn) REFERENCES Technicien_emp,
    FOREIGN KEY (model_no) REFERENCES Model);
```

- On ne peut pas mettre la contrainte de participation sans utiliser de CHECK.

```
CREATE TABLE Plane_type(reg_no INTEGER,
    Model_no INTEGER NOT NULL,
    PRIMARY KEY (reg_no),
    FOREIGN KEY (model_no) REFERENCES Model);
```

```
CREATE TABLE Test_type(FAA_no INTEGER,
    Nom CHAR(20);
    Max_score INTEGER,
    PRIMARY KEY (FAA_no));
```

```
CREATE TABLE TestOf(FFA_no INTEGER,
    Reg_no INTEGER,
    Hours INTEGER,
    Date DATE,
    Score INTEGER,
    Ssn CHAR(9),
    PRIMARY KEY (ssn,reg_no,FFA_no),
    FOREIGN KEY (reg_no) REFERENCES Plane_type,
    FOREIGN KEY (FAA_no) REFERENCES test_type,
    FOREIGN KEY (ssn) REFERENCES Technicien_emp);
```

- On ne peut pas vérifier qu'un technicien soit un expert d'un modèle spécifique s'il fait un test sans utiliser de CHECK CONSTRAINT. Celle-ci serait comme suit si l'on pouvait :

CONSTRAINT MODEL

CHECK (SELECT * FROM Expert, Type

WHERE Expert.ssn = ssn AND

Expert.model_no = Type.model_no AND Type.reg_no = reg_no)