Noé Zwissig analyse

Question

Est-ce que tous les élève au sein d'un classe on FORCEMENT les même leçon ?

OUI

Quel info voulez-vous pour les élève ?

horaire par semestre nom prénom adresse email téléphone statut -> en cours, terminé, abandonné

Quel info voulez-vous pour les enseignants ?

planning nom prénom adresse email téléphone statut -> retraite,pas de cours, en fonction iban

Quel info voulez-vous pour les cours ?

nom description semestre ou il est donnée (évolutif) horaire note semestre

Quel info pour les notes ?

nom description Date de réalisation

Avez-vous des cas spéciaux que vous voulez que nous tenions compte ?

aucun

Des semaines spécial ?

non

Des appuis ?

non aucun

Des fonctionnalité que vous voulez que la future application soit capable de faire ?

aucune

Seulement 3 Entité ? par de salle, bâtiment ou autre ?

salle -> tout les cours sont dans la même salle pour une classe (peut changer)

Voulez-vous un historique ?

oui

Comment fonctionne les promotion ? trimestre semestre libre ?

par trimestre et moyenne de toute les notes supérieur a 4 et module validé

Est-ce que vous voulez que les conditions de promotion soie stockée dans la db ?

oui stocké les fonction et garder un historique

Est-ce que le prof est déterminer par la branche ou par la leçon ?

toujours le même prof par branche mais peut changé année après année

Voulez-vous pouvoir attribué certaine chose à une période spécifique ? (devoir)

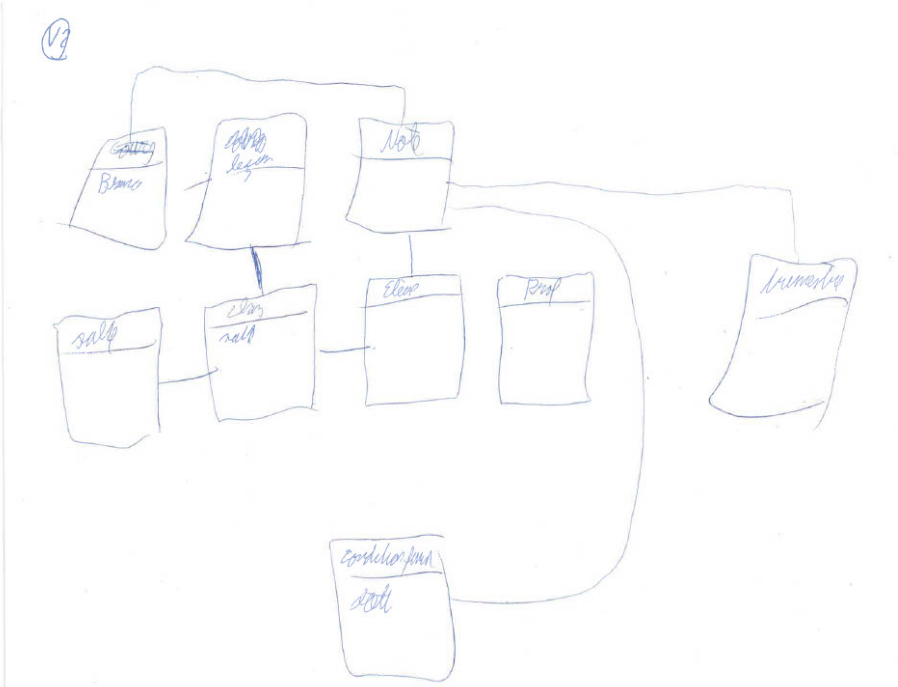
NON

Note de discusion :

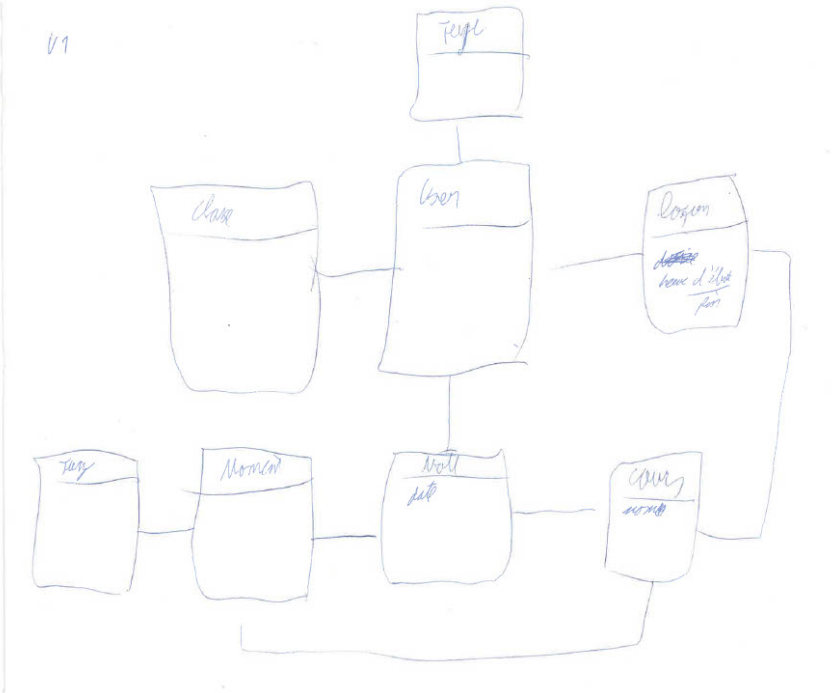
l'horaire sera générer chaque trimestre est parfaitement identique sur toute les semaine qui le compose. (semaine type) un élève peut devenir enseignant

First mind idea on paper

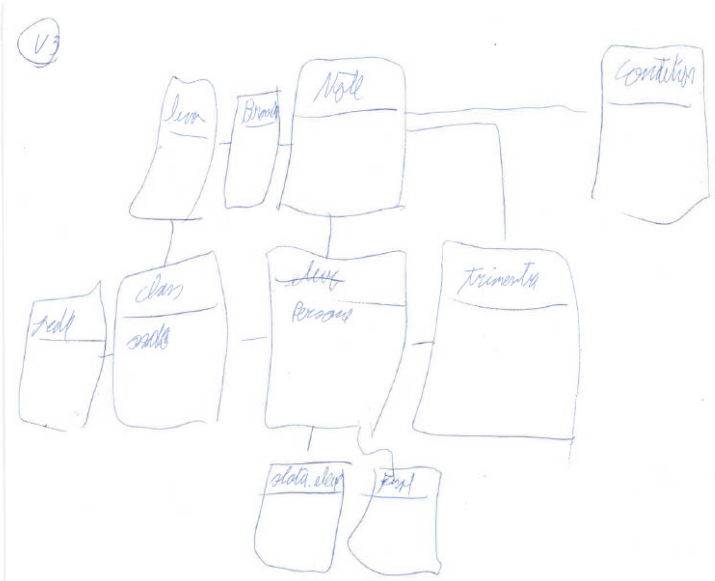
V1



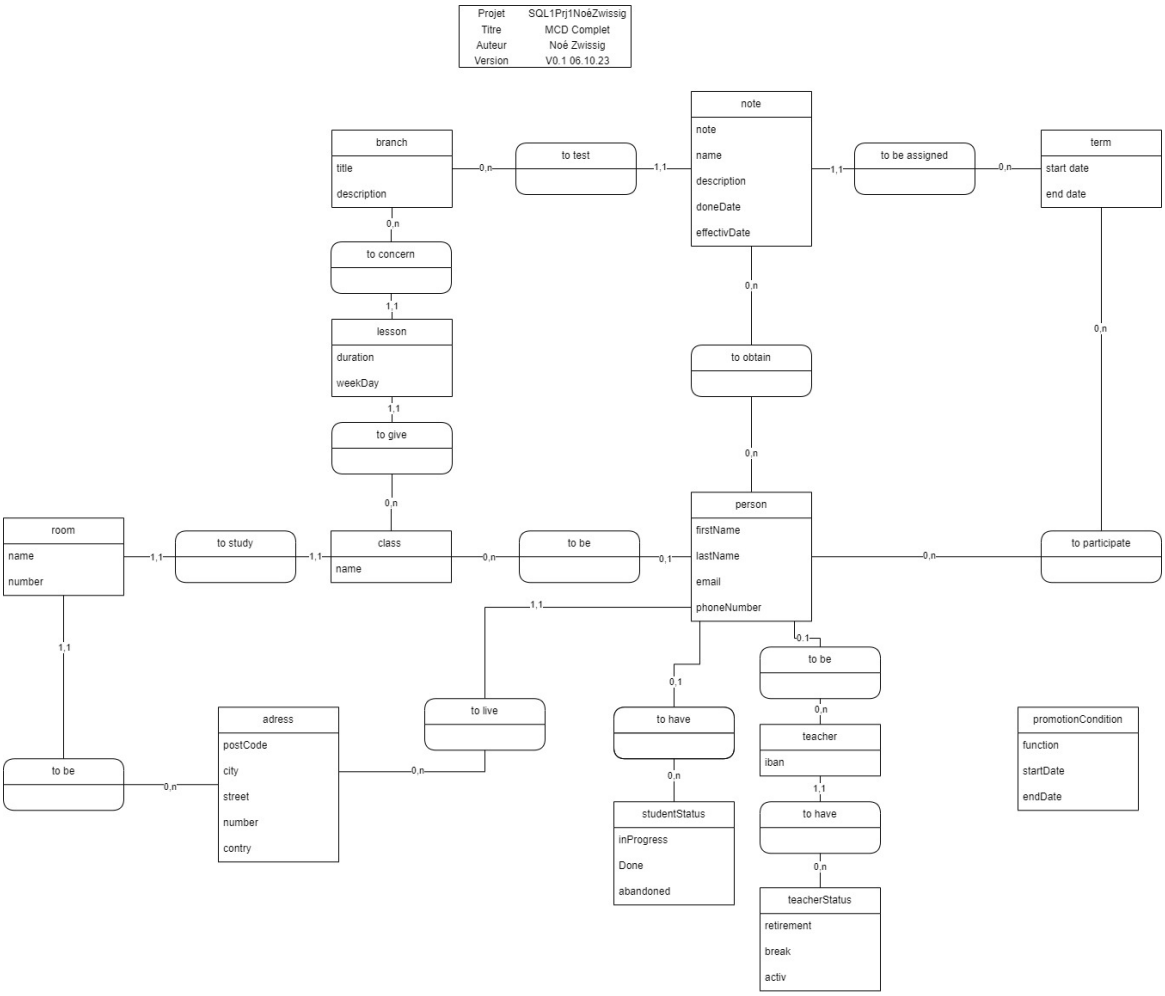
V2



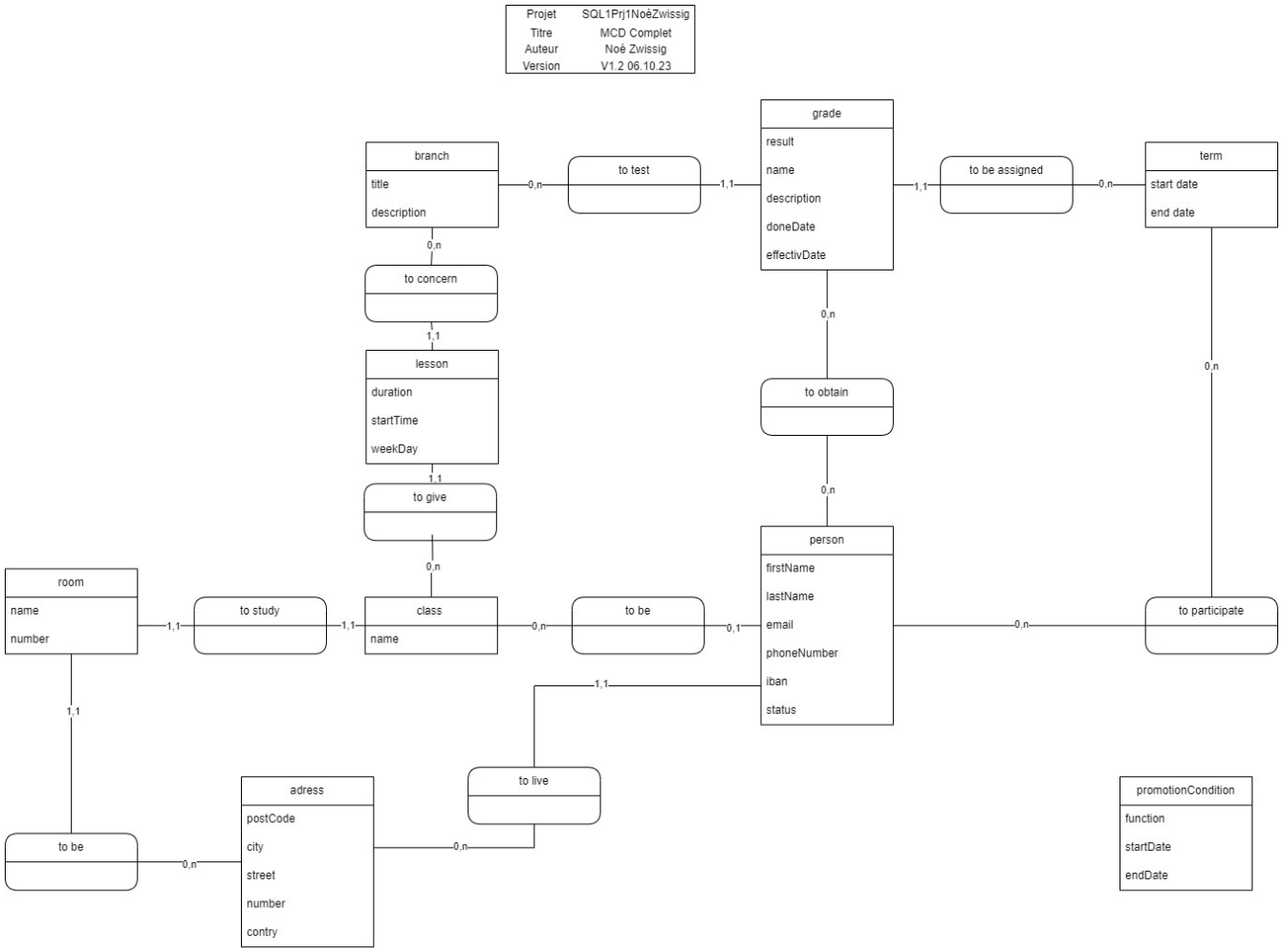
V3



MCD V1.0

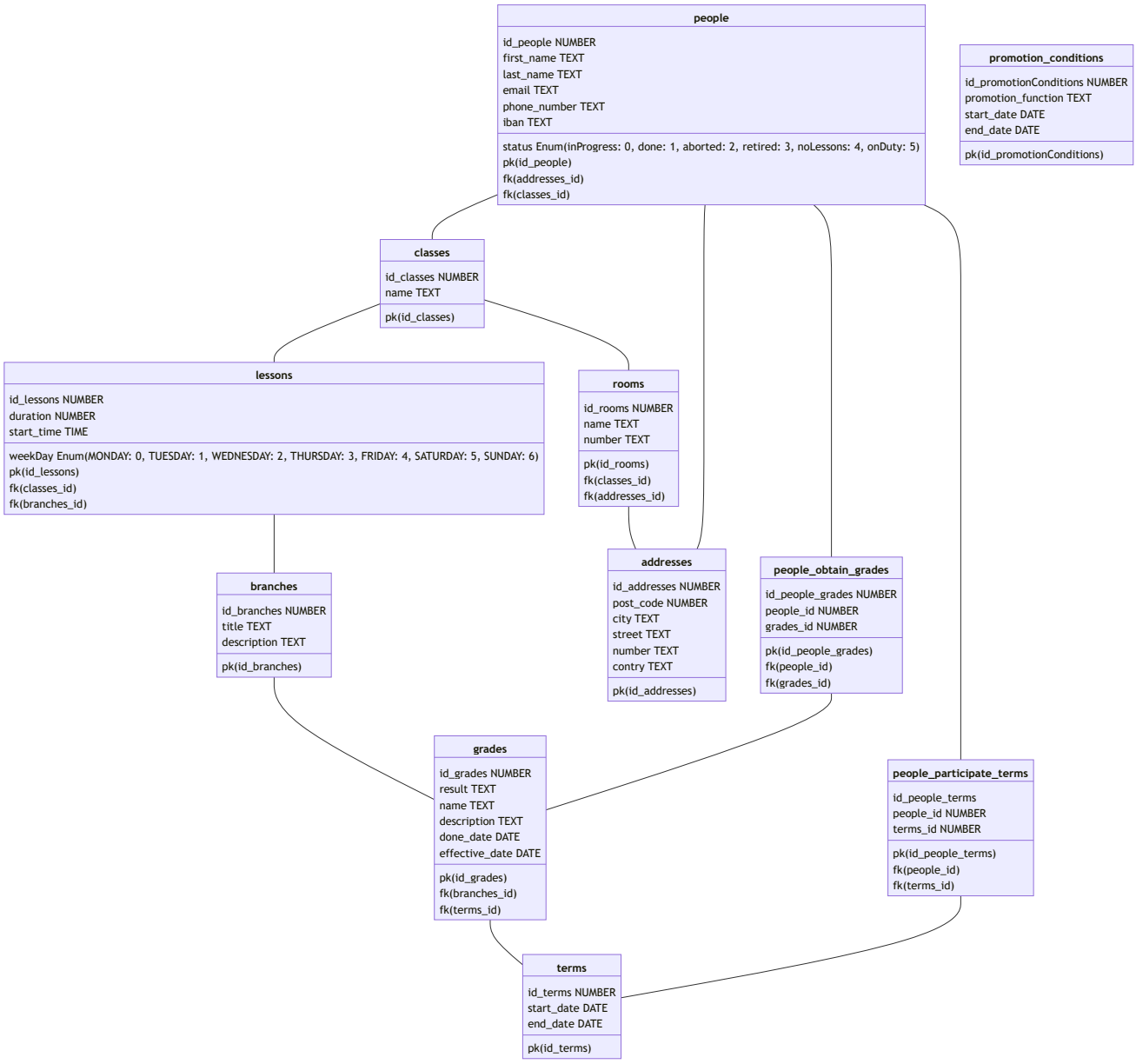

Projet SQL1Prj1NoëZwissig
Titre MCD Complet
Auteur Noé Zwissig
Version V0.1 06.10.23
note
note
name
description
doneDate
effectivDate
branch
title
description
to test
to be assigned
term
start date
end date
to concern
lesson
duration
weekDay
to give
to obtain
person
firstName
lastName
email
phoneNumber
room
name
number
to study
class
name
to be
to participate
to live
adress
postCode
city
street
number
contry
to be
to have
studentStatus
inProgress
Done
abandoned
to be
teacher
iban
to have
teacherStatus
retirement
break
activ
promotionCondition
function
startDate
endDate

MCD V2.0



MLD

In this MLD something as been upgraded. The tables are now in plural. Some typo as also been corrected.



fix inside the v2:

Update the person table to add the status enum inside.
Correct the "note" table as "grade" and the "note" as result.
add startTime in lesson change function to promotion_function because it will generate problem in the creation script.\

Decision and explanation

promotion_conditions

We have made the choice to put the promotion conditions in a separted table.
This is because the promotion conditions are not directly related to a table but need to be stored in the db.\

people

People is the center table of the db.
We have made the choice to store the status directly in an enum inside that table.
From that status we can know if the person is a student or a teacher.
The iban is nullable as only the teacher need to have one.\

classes

A simple table that store all the student together.
It's from that table that we will link the students to the lesson.\

rooms

A simple room table that store the name and the number of the room of a class.\

lessons

We will use the lessons table to generate the schedule.
You can know the day, the start_time and the duration of the lesson.
From that we can generate the schedule for a week.\

branches

The branches table is used to have a name for the lesson or for the grade.\

grades

In the grades table we hava a done_date and a effective_date.
This can be used to find the grades of a student that as been done during a specific period.\

terms

The terms table is used to store the start and end date of a term.
This can be used to know in which term a student is.

This can also be used to know the term of a grade.
We can use that table to research a lot of think. (Ex: find all the grades of a student during a specific term)\

addresses

The addresses table is used to store the address of a person or a room.\

How it works

Each year the admin have to creat a new terms and assign the people to it.
He also need to update the person with the correct class.

By using the database that way, you are able to have an history of who was in wich terms.

SQL script to create and insert data

```sql
CREATE DATABASE IF NOT EXISTS sql1;
USE sql1;

-- Create the "terms" table (No foreign keys)
CREATE TABLE terms (
    id_terms INT PRIMARY KEY,
    start_date DATE,
    end_date DATE
);

-- Create the "branches" table (No foreign keys)
CREATE TABLE branches (
    id_branches INT PRIMARY KEY,
    title TEXT,
    description TEXT
);

-- Create the "grades" table (References "branches" and "terms")
CREATE TABLE grades (
    id_grades INT PRIMARY KEY,
    result TEXT,
    name TEXT,
    description TEXT,
    done_date DATE,
    effective_date DATE,
    branches_id INT,
    terms_id INT,
    FOREIGN KEY (branches_id) REFERENCES branches (id_branches),
    FOREIGN KEY (terms_id) REFERENCES terms (id_terms)
);
```

```sql
-- Create the "classes" table (No foreign keys)
CREATE TABLE classes (
    id_classes INT PRIMARY KEY,
    name TEXT
);

-- Create the "addresses" table (No foreign keys)
CREATE TABLE addresses (
    id_addresses INT PRIMARY KEY,
    post_code INT,
    city TEXT,
    street TEXT,
    number TEXT,
    country TEXT
);

-- Create the "rooms" table (References "classes" and "addresses")
CREATE TABLE rooms (
    id_rooms INT PRIMARY KEY,
    name TEXT,
    number TEXT,
    classes_id INT,
    addresses_id INT,
    FOREIGN KEY (classes_id) REFERENCES classes (id_classes),
    FOREIGN KEY (addresses_id) REFERENCES addresses (id_addresses)
);

-- Create the "people" table (References "addresses" and "classes")
CREATE TABLE people (
    id_people INT PRIMARY KEY,
    first_name TEXT,
    last_name TEXT,
    email TEXT,
    phone_number TEXT,
    iban TEXT,
    status ENUM('inProgress', 'done', 'aborted', 'retired', 'noLessons', 'onDuty'),
    addresses_id INT,
    classes_id INT,
    FOREIGN KEY (addresses_id) REFERENCES addresses (id_addresses),
    FOREIGN KEY (classes_id) REFERENCES classes (id_classes)
);

-- Create the "people_obtain_grades" table (References "people" and "grades")
CREATE TABLE people_obtain_grades (
    id_people_grades INT PRIMARY KEY,
    people_id INT,
    grades_id INT,
    FOREIGN KEY (people_id) REFERENCES people (id_people),
    FOREIGN KEY (grades_id) REFERENCES grades (id_grades)
);

-- Create the "people_participate_terms" table (References "people" and "terms")
CREATE TABLE people_participate_terms (
    id_people_terms INT PRIMARY KEY,
    people_id INT,
    terms_id INT,
```

```
    FOREIGN KEY (people_id) REFERENCES people (id_people),
    FOREIGN KEY (terms_id) REFERENCES terms (id_terms)
);

-- Create the "promotion_conditions" table (No foreign keys)
CREATE TABLE promotion_conditions (
    id_promotionConditions INT PRIMARY KEY,
    promotion_function TEXT,
    start_date DATE,
    end_date DATE
);

-- Create the "lessons" table (References "classes" and "branches")
CREATE TABLE lessons (
    id_lessons INT PRIMARY KEY,
    duration INT,
    start_time TIME,
    weekDay ENUM('MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY'),
    classes_id INT,
    branches_id INT,
    FOREIGN KEY (classes_id) REFERENCES classes (id_classes),
    FOREIGN KEY (branches_id) REFERENCES branches (id_branches)
);

-- Insert data into the "terms" table
INSERT INTO terms (id_terms, start_date, end_date)
VALUES
    (1, '2023-01-15', '2023-06-30'),
    (2, '2023-07-01', '2023-12-31'),
    (3, '2023-01-15', '2023-06-30'),
    (4, '2023-07-01', '2023-12-31'),
    (5, '2023-01-15', '2023-06-30');

-- Insert data into the "branches" table
INSERT INTO branches (id_branches, title, description)
VALUES
    (1, 'Mathematics', 'Study of numbers and shapes'),
    (2, 'English', 'Language and literature studies'),
    (3, 'Engineering', 'Design and problem-solving'),
    (4, 'Business', 'Economics and management'),
    (5, 'Health Sciences', 'Medical and health-related fields');

-- Insert data into the "grades" table
INSERT INTO grades (id_grades, result, name, description, done_date, effective_date, branches_id, terms_
VALUES
    (1, '1.0', 'F', 'Fail', '2023-05-15', '2023-06-30', 1, 1),
    (2, '2.0', 'D', 'Needs improvement', '2023-06-30', '2023-12-31', 2, 2),
    (3, '2.5', 'C', 'Satisfactory', '2023-06-30', '2023-12-31', 1, 2),
    (4, '3.0', 'B', 'Good performance', '2023-01-15', '2023-06-30', 3, 1),
    (5, '3.5', 'B+', 'Very good', '2023-07-01', '2023-12-31', 4, 2);

-- Insert data into the "classes" table
INSERT INTO classes (id_classes, name)
VALUES
    (1, 'Class A'),
    (2, 'Class B'),
    (3, 'Class C'),
```

```
    (4, 'Class D'),
    (5, 'Class E');

-- Insert data into the "addresses" table
INSERT INTO addresses (id_addresses, post_code, city, street, number, country)
VALUES
    (1, 12345, 'New York', 'Main St', '123', 'USA'),
    (2, 54321, 'Los Angeles', 'Oak Ave', '456', 'USA'),
    (3, 98765, 'London', 'Baker St', '789', 'UK'),
    (4, 45678, 'Paris', 'Champs Elysées', '987', 'France'),
    (5, 56789, 'Berlin', 'Brandenburg Strasse', '654', 'Germany');

-- Insert data into the "rooms" table
INSERT INTO rooms (id_rooms, name, number, classes_id, addresses_id)
VALUES
    (1, 'Room 101', '101A', 1, 1),
    (2, 'Room 201', '201B', 2, 2),
    (3, 'Room 301', '301C', 3, 3),
    (4, 'Room 401', '401D', 4, 4),
    (5, 'Room 501', '501E', 5, 5);

-- Insert data into the "people" table
INSERT INTO people (id_people, first_name, last_name, email, phone_number, iban, status, addresses_id, c
VALUES
    (1, 'John', 'Doe', 'john.doe@email.com', '+123456789', 'US123456789', 'done', 1, 1),
    (2, 'Jane', 'Smith', 'jane.smith@email.com', '+987654321', 'UK987654321', 'inProgress', 2, 2),
    (3, 'Robert', 'Johnson', 'robert.johnson@email.com', '+1122334455', 'FR1122334455', 'done', 3, 3),
    (4, 'Ella', 'Wilson', 'ella.wilson@email.com', '+5544332211', 'DE5544332211', 'onDuty', 4, 4),
    (5, 'Michael', 'Brown', 'michael.brown@email.com', '+2233445566', 'US2233445566', 'aborted', 5, 5);

-- Insert data into the "people_obtain_grades" table
INSERT INTO people_obtain_grades (id_people_grades, people_id, grades_id)
VALUES
    (1, 1, 1),
    (2, 2, 2),
    (3, 3, 3),
    (4, 4, 4),
    (5, 5, 5);

-- Insert data into the "people_participate_terms" table
INSERT INTO people_participate_terms (id_people_terms, people_id, terms_id)
VALUES
    (1, 1, 1),
    (2, 2, 2),
    (3, 3, 3),
    (4, 4, 4),
    (5, 5, 5);

-- Insert data into the "promotion_conditions" table
INSERT INTO promotion_conditions (id_promotionConditions, promotion_function, start_date, end_date)
VALUES
    (1, 'Function A', '2023-01-01', '2023-12-31'),
    (2, 'Function B', '2023-04-01', '2023-09-30'),
    (3, 'Function C', '2023-02-15', '2023-08-15'),
    (4, 'Function D', '2023-06-01', '2023-11-30'),
    (5, 'Function E', '2023-03-01', '2023-08-31');
```

```sql
-- Insert data into the "lessons" table
INSERT INTO lessons (id_lessons, duration, start_time, weekDay, classes_id, branches_id)
VALUES
    (1, 90, '08:00:00', 'MONDAY', 1, 1),
    (2, 75, '09:30:00', 'TUESDAY', 2, 2),
    (3, 120, '10:45:00', 'WEDNESDAY', 3, 3),
    (4, 60, '13:15:00', 'THURSDAY', 4, 4),
    (5, 105, '14:30:00', 'FRIDAY', 5, 5);
```