5) Specification of database in SQL

```
-- Student Table with optional attribute Work

CREATE TABLE Student (

   StudentID SERIAL PRIMARY KEY,

   FirstName VARCHAR(50) NOT NULL,

   LastName VARCHAR(50) NOT NULL,

   DateOfBirth DATE NOT NULL,

   Email VARCHAR(100),

   Level VARCHAR(50) NOT NULL,

   Work VARCHAR(255) -- Optional attribute

);


-- Professor Table with optional attributes Department and Affiliation

CREATE TABLE Professor (

   ProfessorID SERIAL PRIMARY KEY,

   FirstName VARCHAR(50) NOT NULL,

   LastName VARCHAR(50) NOT NULL,

   Email VARCHAR(100),

   Department VARCHAR(100), -- Optional attribute for UniversityProfessor

   Affiliation VARCHAR(100) -- Optional attribute for ExternalProfessor

);


-- Phone Table with both attributes as primary key

CREATE TABLE Phone (

   Number VARCHAR(15) NOT NULL,

   ProfessorID INTEGER NOT NULL,
    PRIMARY KEY (Number, ProfessorID),
    FOREIGN KEY (ProfessorID) REFERENCES Professor(ProfessorID)
);

-- Classroom Table with composite primary key (RoomNr, Building)
CREATE TABLE Classroom (
    RoomNr VARCHAR(10) NOT NULL,
    Building VARCHAR(50) NOT NULL,
    Capacity INTEGER NOT NULL CHECK (Capacity >= 0),
    PRIMARY KEY (RoomNr, Building)
);

-- Exam Table with foreign keys to Professor, Subject, and Classroom
CREATE TABLE Exam (
    ExamID SERIAL PRIMARY KEY,
    ExamDate DATE NOT NULL,
    ExamStartTime TIME NOT NULL,
    ExamEndTime TIME NOT NULL,
    ProfessorID INTEGER NOT NULL,
    SubjectID INTEGER NOT NULL,
    RoomNr VARCHAR(10) NOT NULL,
    Building VARCHAR(50) NOT NULL,
    FOREIGN KEY (ProfessorID) REFERENCES Professor(ProfessorID),
    FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID),
    FOREIGN KEY (RoomNr, Building) REFERENCES Classroom(RoomNr, Building)
);
```

```
-- Subject Table
CREATE TABLE Subject (
   SubjectID SERIAL,
   ProfessorID INTEGER NOT NULL,
   SubjectName VARCHAR(50) NOT NULL,
   PRIMARY KEY (SubjectID, ProfessorID),
   FOREIGN KEY (ProfessorID) REFERENCES Professor(ProfessorID)
);
-- TaughtBy Table
CREATE TABLE TaughtBy (
   SubjectID INTEGER NOT NULL,
   ProfessorID INTEGER NOT NULL,
   PRIMARY KEY (SubjectID, ProfessorID),
   FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID),
   FOREIGN KEY (ProfessorID) REFERENCES Professor(ProfessorID)
);

-- Takes Table with inclusion constraint
CREATE TABLE Takes (
   StudentID INTEGER NOT NULL,
   ExamID INTEGER NOT NULL,
   PRIMARY KEY (StudentID, ExamID),
   FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
   FOREIGN KEY (ExamID) REFERENCES Exam(ExamID)
);
```

```sql
-- Hosts Table with composite primary key
CREATE TABLE Hosts (
    RoomNr VARCHAR(10) NOT NULL,
    Building VARCHAR(50) NOT NULL,
    SubjectID INTEGER NOT NULL,
    PRIMARY KEY (RoomNr, Building, SubjectID),
    FOREIGN KEY (RoomNr, Building) REFERENCES Classroom(RoomNr, Building),
    FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
);

-- Mentors Table
CREATE TABLE Mentors (
    ProfessorID INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
    PRIMARY KEY (ProfessorID, StudentID),
    FOREIGN KEY (ProfessorID) REFERENCES Professor(ProfessorID),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
);
```