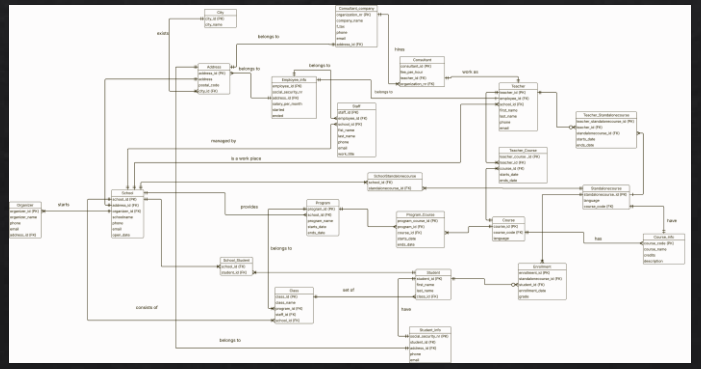
YrkesCo

Lite om mig

Kort om vad presentation kommer handla om



Mjukvaror

Luicidchart

dbdiagram

postgreSQL

docker

visual studio code

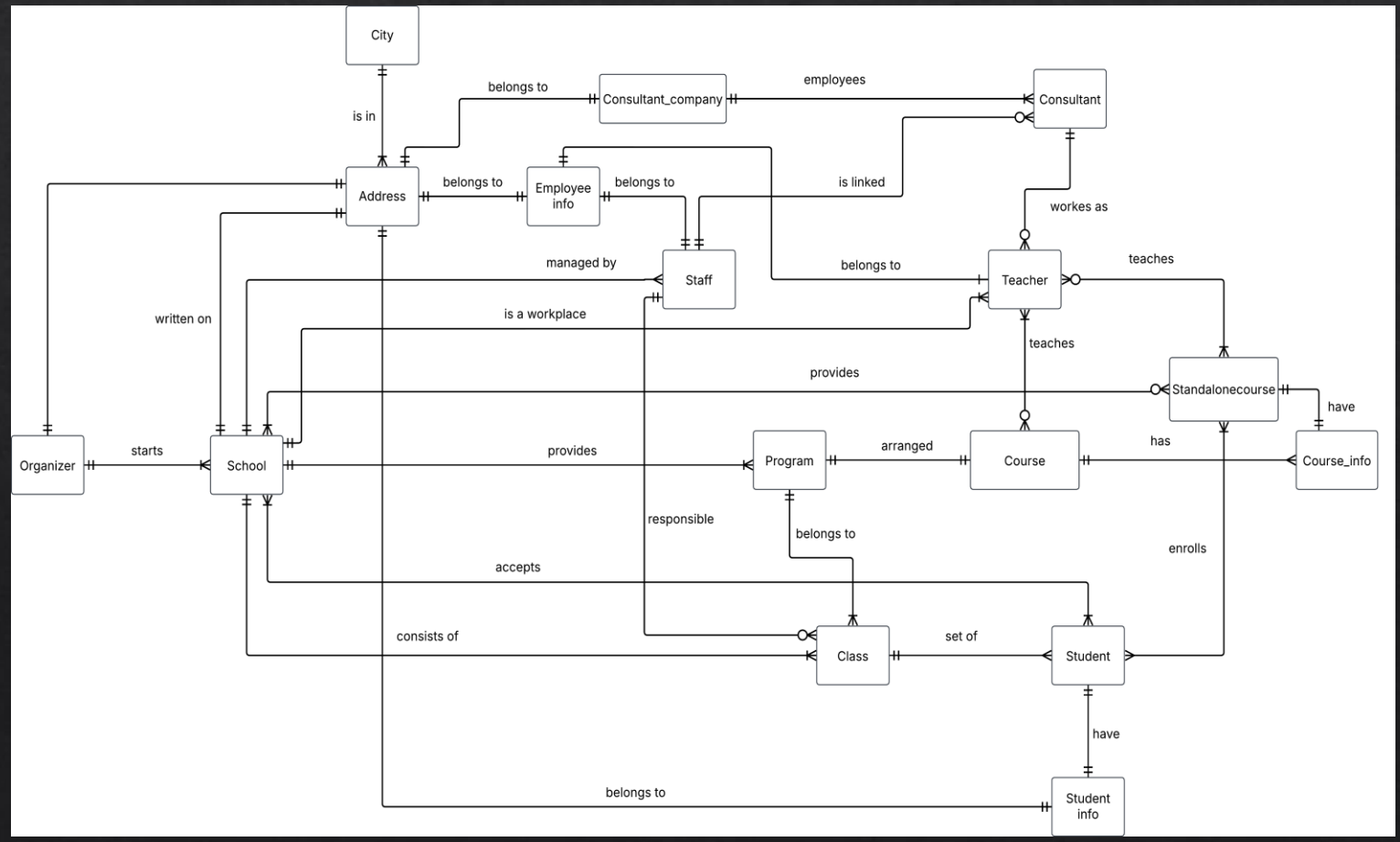
Conceptual model

Relationer mellan entiteterna

- Strukturen
- Kardinalitet
- Relationship statement

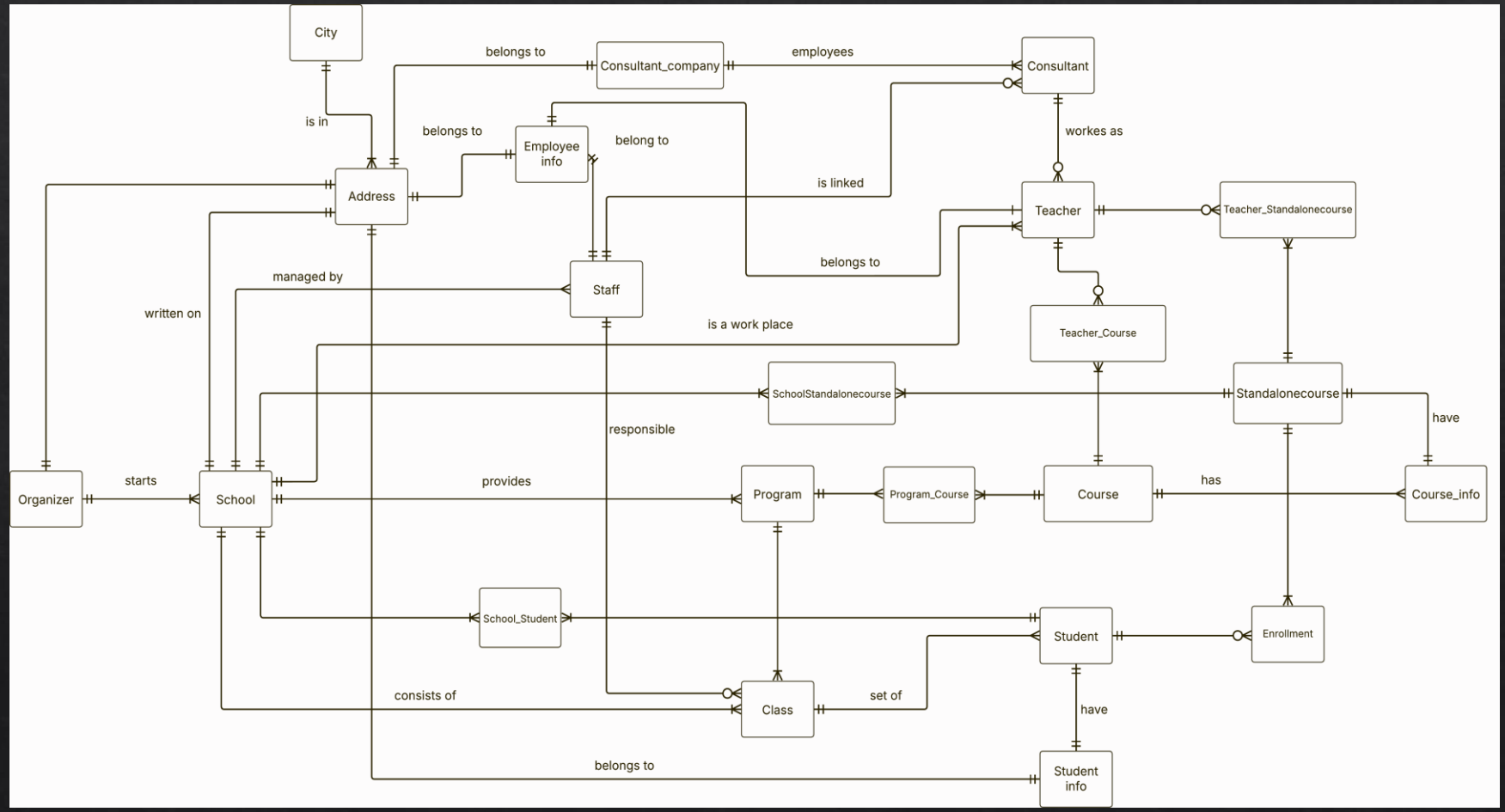
Business rules

- Ett program kan bara existera på en skola.
- Ett program har alltid en klass.
- En klass kan ha noll studenter
- Man behöver inte tillhöra en klass för att skriva in sig på en fristående kurs.
- En student kan bara tillhöra en klass.
- Personal kan bara tillhöra en skola
- Utbildningsledare måste vara anställd



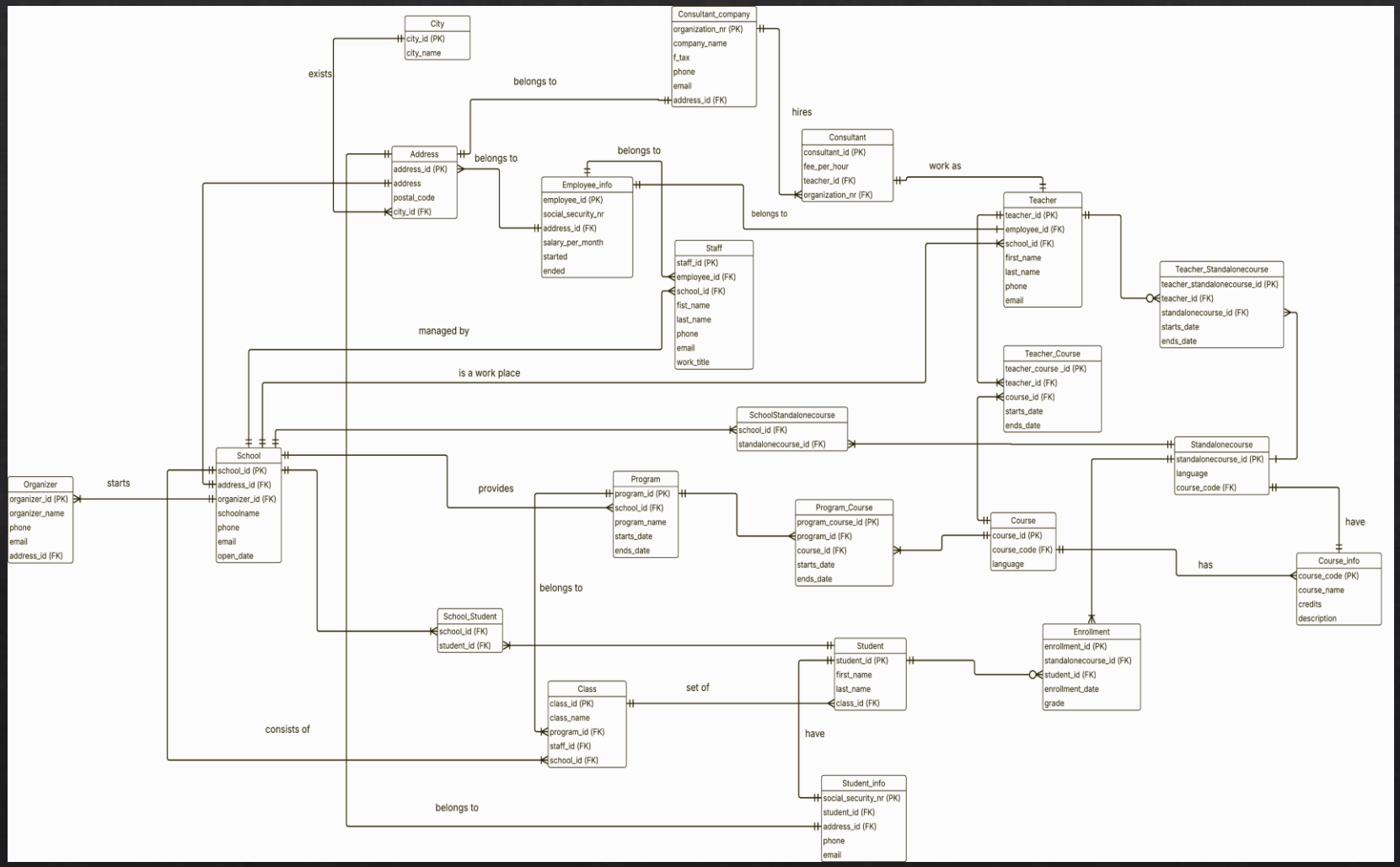
Composite Entity

- Vilka som är compoiste entity
- Varför man måste använda dem



Logical ERD

- Attribut
- Primary key
- Foregein key
- Hantera olika anomalier
- Normalisering
- 1NF
- 2NF
- 3NF



Physical model

- Domain
- Domain constraint
- Typ av databas

```sql
CREATE TABLE IF NOT EXISTS School (
    school_id SERIAL PRIMARY KEY,
    organizer_id INTEGER NOT NULL,
    address_id INTEGER NOT NULL,
    schoolname VARCHAR(100) NOT NULL,
    phone  VARCHAR(20) UNIQUE NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL CHECK (email LIKE '%@%'),
    open_date DATE NOT NULL,
    FOREIGN KEY (address_id) REFERENCES Address (address_id) ON DELETE CASCADE,
    FOREIGN KEY (organizer_id) REFERENCES Organizer (organizer_id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS "Program" (
    program_id SERIAL PRIMARY KEY,
    school_id INTEGER NOT NULL,
    program_name VARCHAR(100) NOT NULL,
    starts_date DATE NOT NULL,
    ends_date DATE NOT NULL,
    FOREIGN KEY (school_id) REFERENCES School (school_id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Course_info (
    course_code VARCHAR(25) PRIMARY KEY,
    course_name VARCHAR(50) NOT NULL,
    credits INTEGER NOT NULL,
    descriptions TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS Course (
    course_id SERIAL PRIMARY KEY,
    course_code VARCHAR(25) NOT NULL,
    languages VARCHAR(10) NOT NULL,
    FOREIGN KEY (course_code) REFERENCES Course_info (course_code) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Standalonecourse (
    standalonecourse_id SERIAL PRIMARY KEY,
    languages VARCHAR(10) NOT NULL,
```

⊗ 88 ⚠ 0 Select Postgres Server

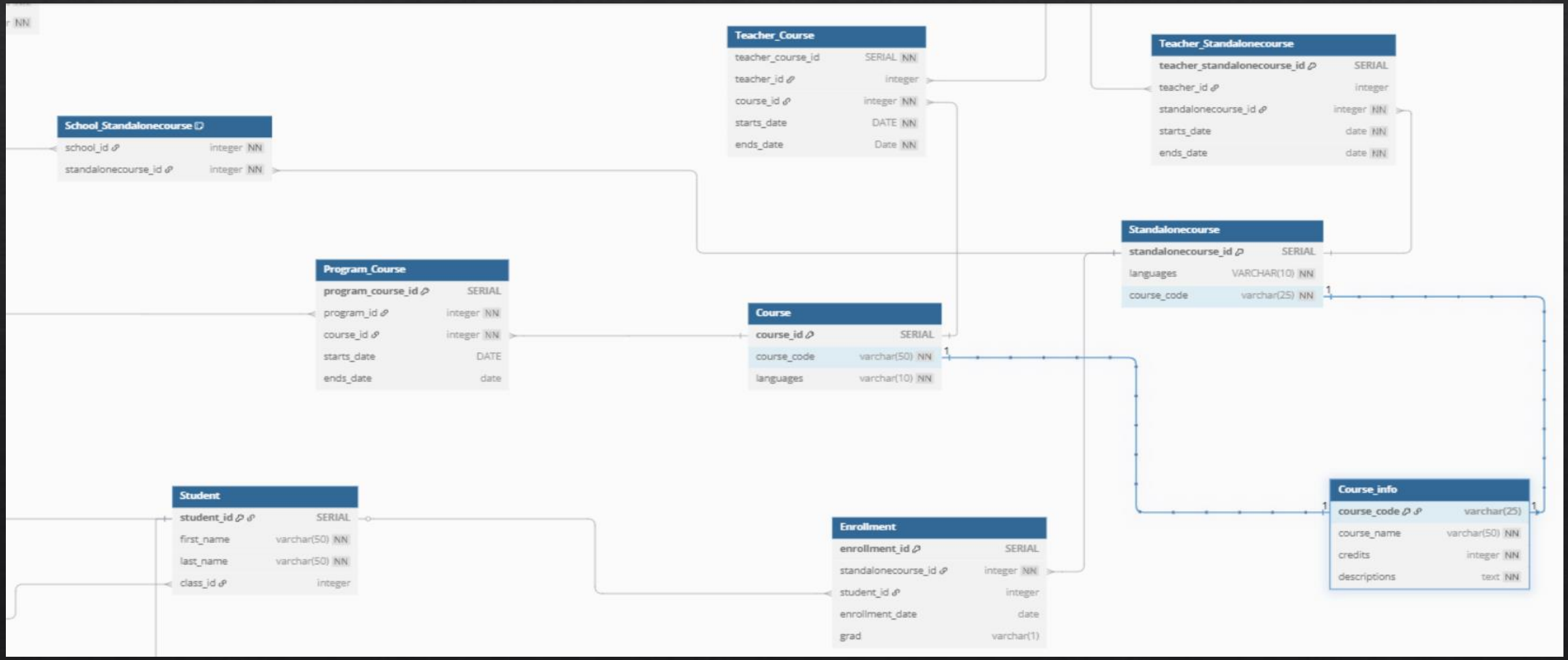
```sql
CREATE TABLE IF NOT EXISTS Teacher_Course (
    teacher_course_id SERIAL PRIMARY KEY,
    teacher_id INTEGER,
    course_id INTEGER NOT NULL,
    starts_date DATE NOT NULL,
    ends_date DATE NOT NULL,
    FOREIGN KEY (teacher_id) REFERENCES Teacher (teacher_id) ON DELETE CASCADE,
    FOREIGN KEY (course_id) REFERENCES Course (course_id) ON DELETE CASCADE
);


CREATE TABLE IF NOT EXISTS Teacher_Standalonecourse (
    teacher_standalonecourse_id SERIAL PRIMARY KEY,
    teacher_id INTEGER,
    standalonecourse_id INTEGER NOT NULL,
    starts_date DATE NOT NULL,
    ends_date DATE NOT NULL,
    FOREIGN KEY (teacher_id) REFERENCES Teacher (teacher_id) ON DELETE CASCADE,
    FOREIGN KEY (standalonecourse_id) REFERENCES Standalonecourse (standalonecourse_id) ON DELETE CASCADE
);


CREATE TABLE IF NOT EXISTS School_Standalonecourse (
    school_id INTEGER NOT NULL,
    standalonecourse_id INTEGER NOT NULL,
    PRIMARY KEY (school_id, standalonecourse_id), -- Composite primary key
    FOREIGN KEY (school_id) REFERENCES School (school_id) ON DELETE CASCADE,
    FOREIGN KEY (standalonecourse_id) REFERENCES Standalonecourse (standalonecourse_id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS School_Student (
    school_id INTEGER NOT NULL,
    student_id INTEGER NOT NULL,
    PRIMARY KEY (school_id, student_id), -- composite primary key
    FOREIGN KEY (school_id) REFERENCES School (school_id) ON DELETE CASCADE,
    FOREIGN KEY (student_id) REFERENCES Student (student_id) ON DELETE CASCADE
);
```

DBdiagram


Teacher_Course
teacher_course_id SERIAL NN
teacher_id integer
course_id integer NN
starts_date DATE NN
ends_date Date NN
Teacher_Standalonecourse
teacher_standalonecourse_id SERIAL
teacher_id integer
standalonecourse_id integer NN
starts_date date NN
ends_date date NN
School_Standalonecourse
school_id integer NN
standalonecourse_id integer NN
Standalonecourse
standalonecourse_id SERIAL
languages VARCHAR(10) NN
course_code varchar(25) NN
Program_Course
program_course_id SERIAL
program_id integer NN
course_id integer NN
starts_date DATE
ends_date date
Course
course_id SERIAL
course_code varchar(50) NN
languages varchar(10) NN
Student
student_id SERIAL
first_name varchar(50) NN
last_name varchar(50) NN
class_id integer
Enrollment
enrollment_id SERIAL
standalonecourse_id integer NN
student_id integer
enrollment_date date
grad varchar(1)
Course_info
course_code varchar(25)
course_name varchar(50) NN
credits integer NN
descriptions text NN