John Sandsjö

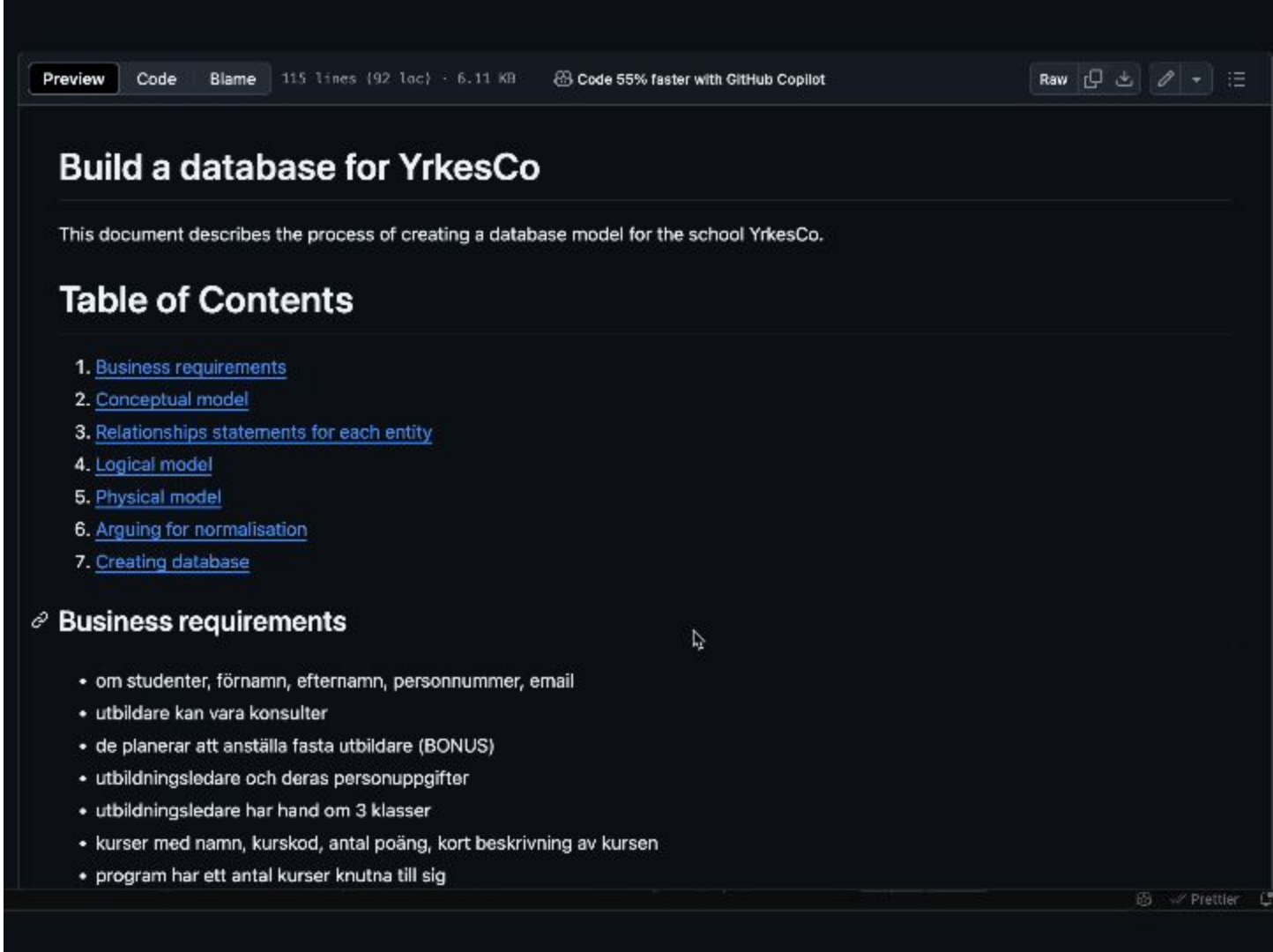
DE24 - Datamodellering

Labb - databas för yrkeshögskolan YrkesCo

Agenda

Background
Business objective
Business requirements
Conceptual model
Logical model
Physical model
The data
Normalisation
Some query results and why they matter to the business

Note
The information in this presentation can also be read in the yh_labb_report.md file in my GitHub repo



Background

YrkesCo is a Swedish school with focus on Data and AI. It currently has two locations, one in Gothenburg and one in Stockholm. The schools program are location specific but the school also offer standalone courses to the students that can be taken from both sites.

Business objectives

Build a scalable database for YrkesCo. The model should adhere to future business needs like being able to scaling the operations to more sites as well as adhering to Personal Identifiable Information (PII).

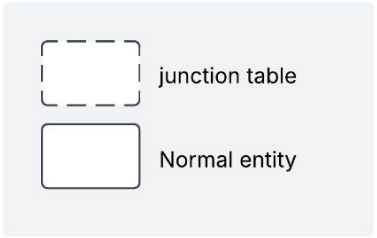
Business requirements, provided in task

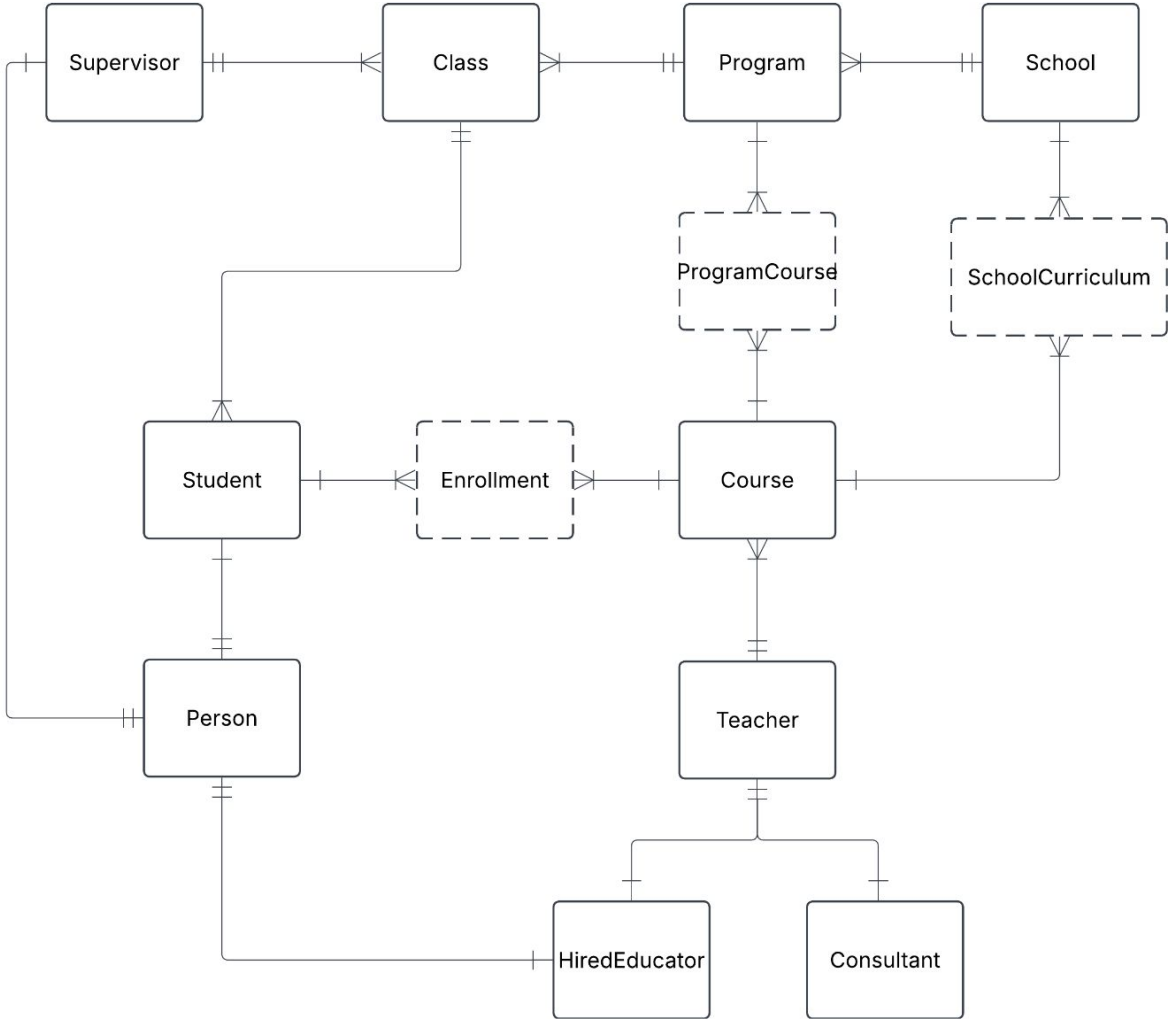
- om studenter, förnamn, efternamn, personnummer, email
- utbildare kan vara konsulter
- de planerar att anställa fasta utbildare (BONUS)
- utbildningsledare och deras personuppgifter
- utbildningsledare har hand om 3 klasser
- kurser med namn, kurskod, antal poäng, kort beskrivning av kursen
- program har ett antal kurser knutna till sig
- ett program blir beviljat i tre omgångar, dvs att det finns 3 klasser
- det finns även fristående kurser (BONUS)
- konsulter, deras företag, företagsinfo som organisationsnummer, har F-skatt address, hur mycket de tar i arvode per timma
- YrkesCo har två anläggningar, en i göteborg och en i stockholm, i framtiden kanske de kommer expandera till flera orter (BONUS)

Additional business requirements

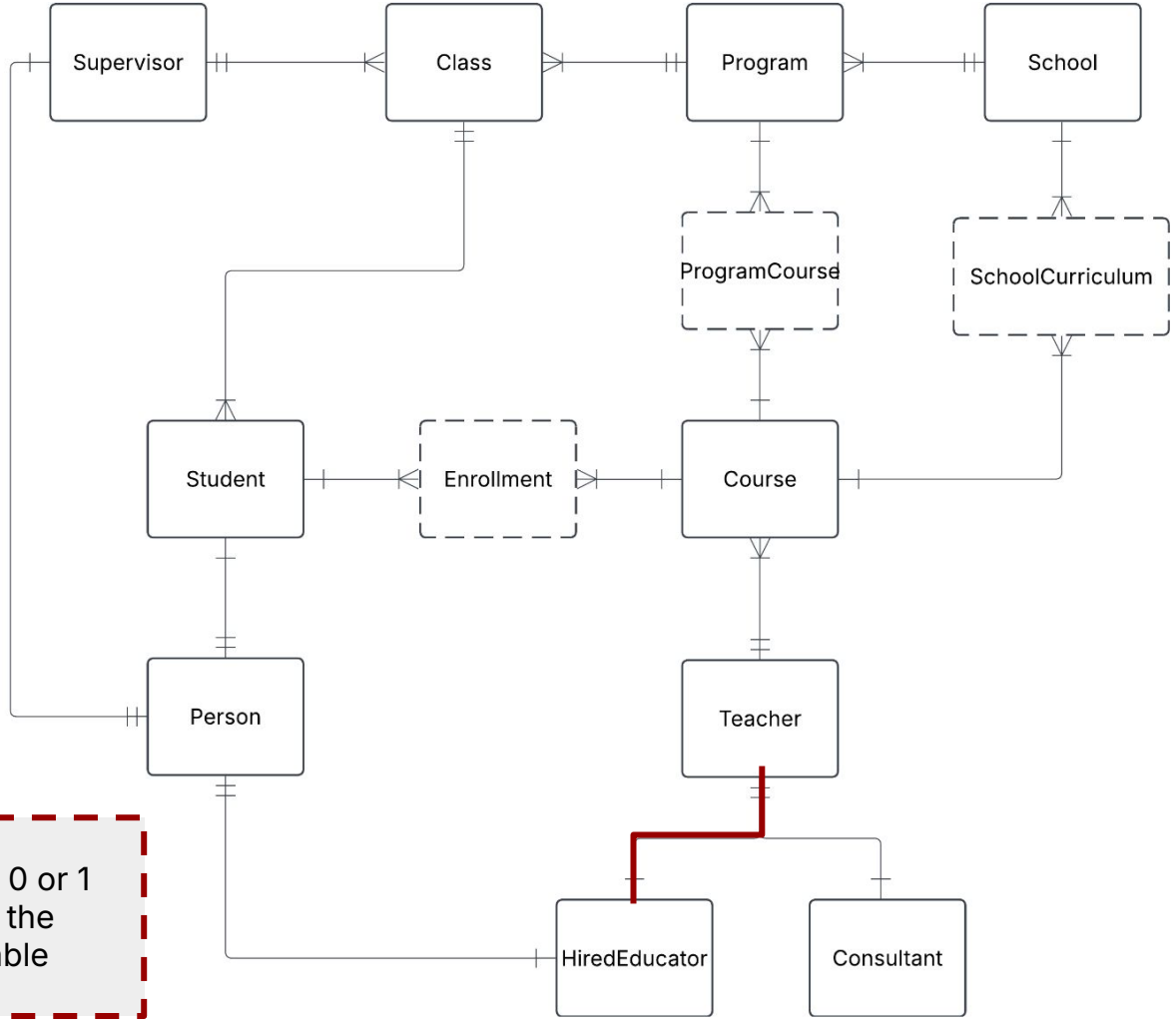
- A course can have only one teacher
- Standalone course can be taken online by student in any of schools. But courses are only open for students enrolled in the school
- A program belongs to one school site
- A course can belong to many programs (e.g. a Python fundamentals course)

Conceptual model





Conceptual model

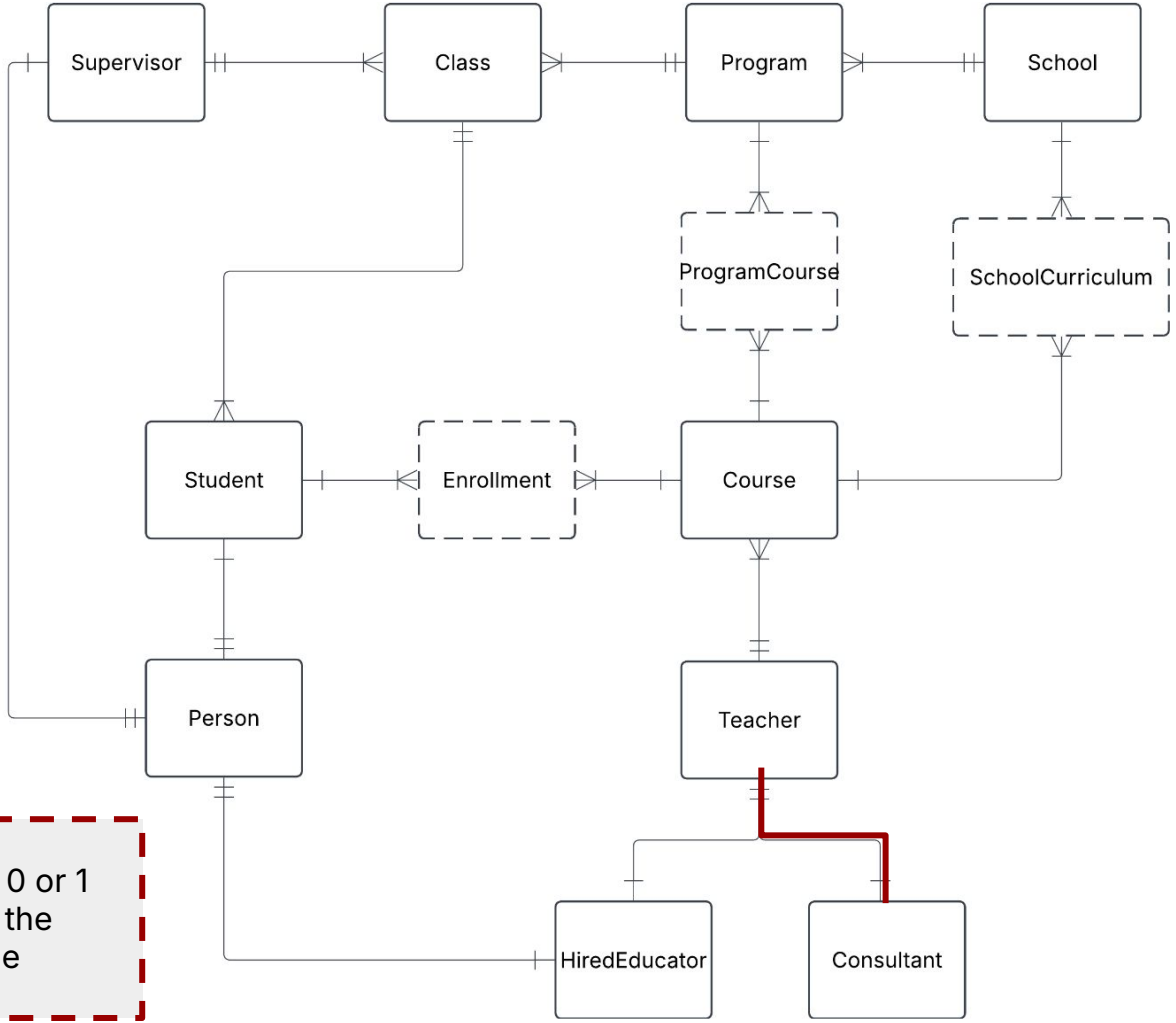


junction table

Normal entity

A teacher can have 0 or 1 representation in the HiredEducator table

Conceptual model

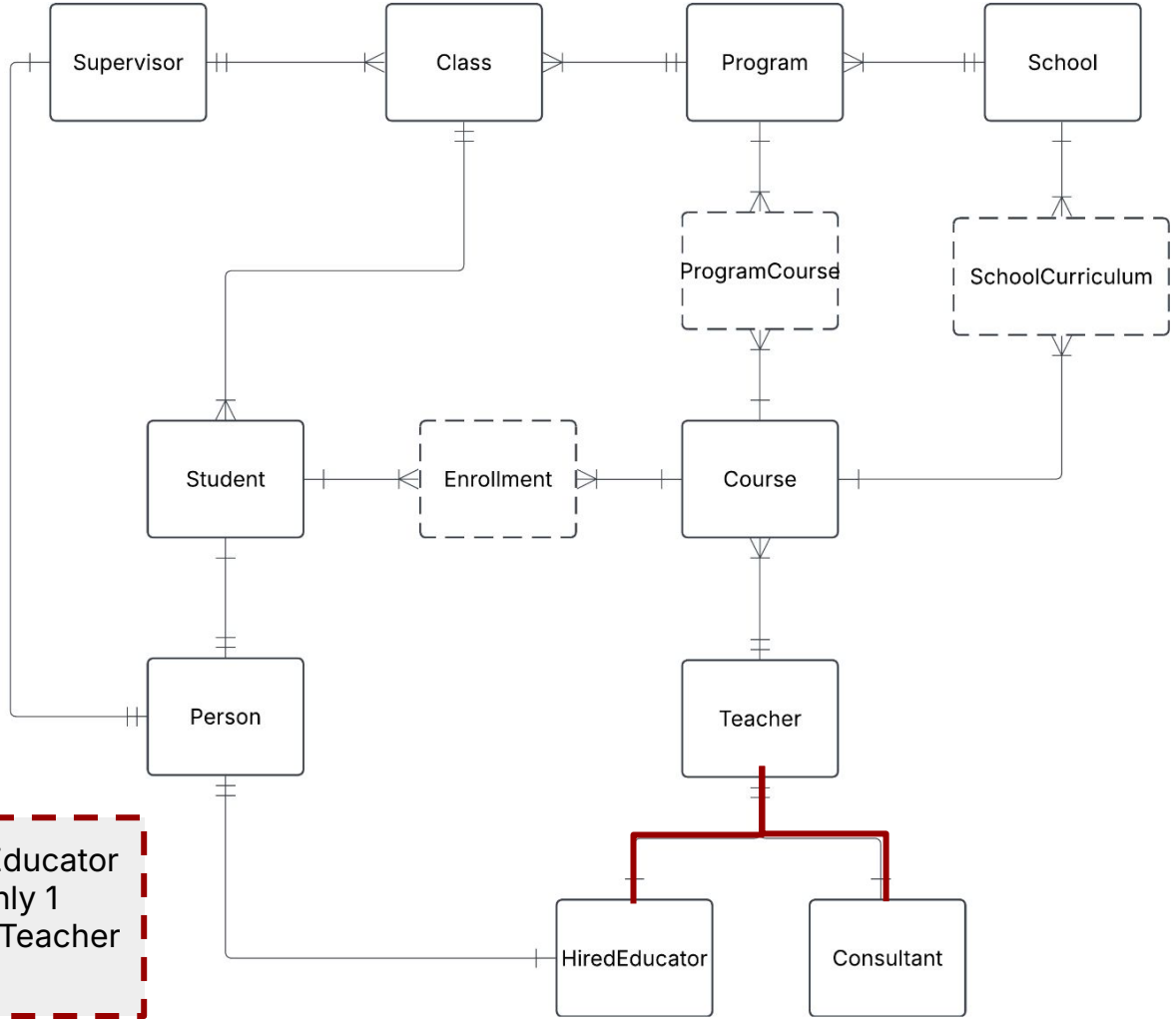


junction table

Normal entity

A teacher can have 0 or 1 representation in the Consultant table

Conceptual model

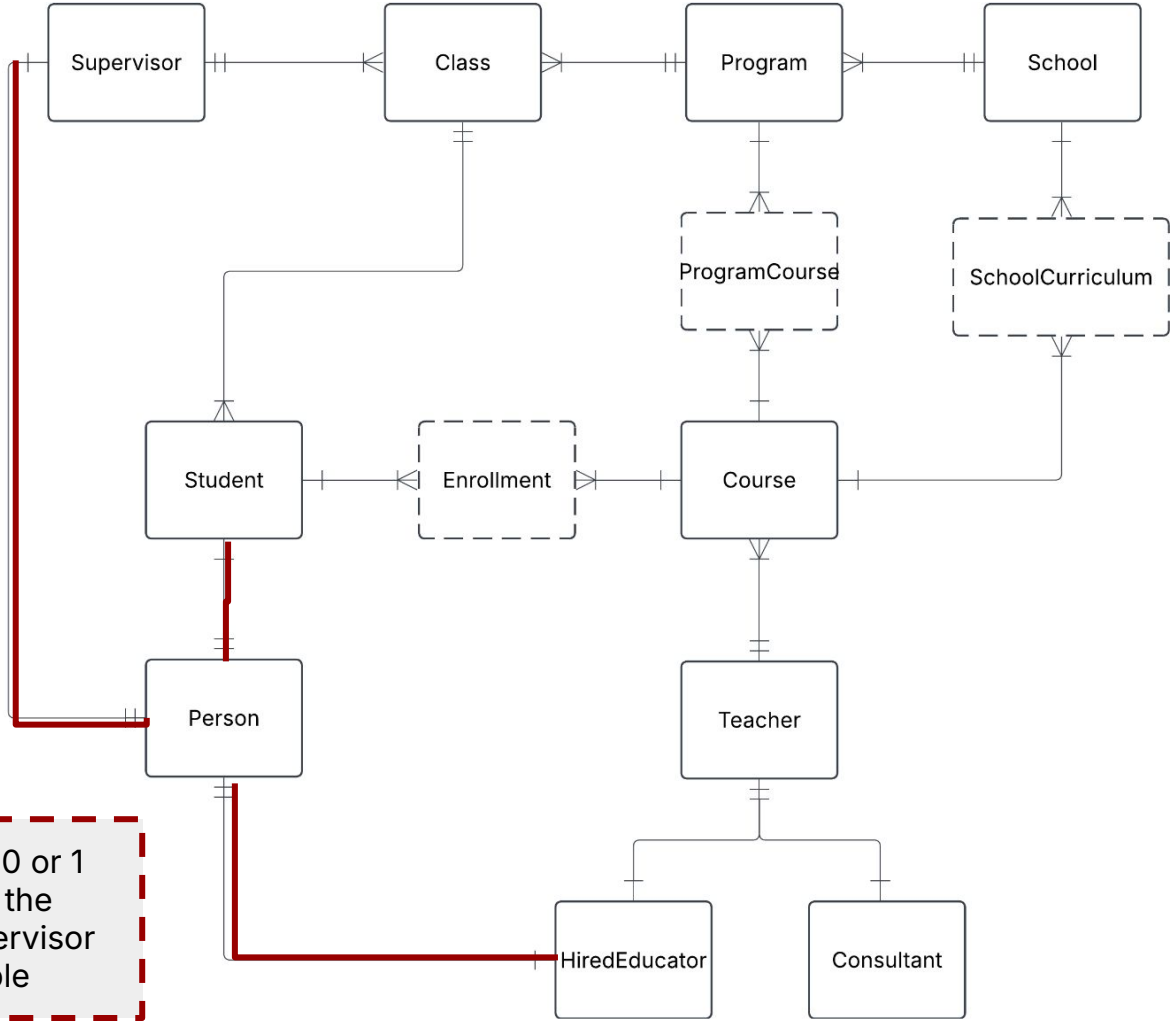


junction table

Normal entity

A Consultant / HiredEducator can have 1 and only 1 representation in the Teacher table

Conceptual model

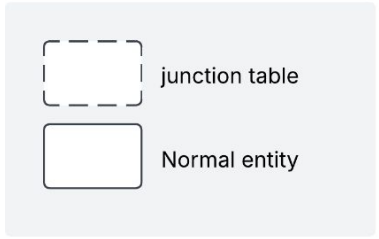


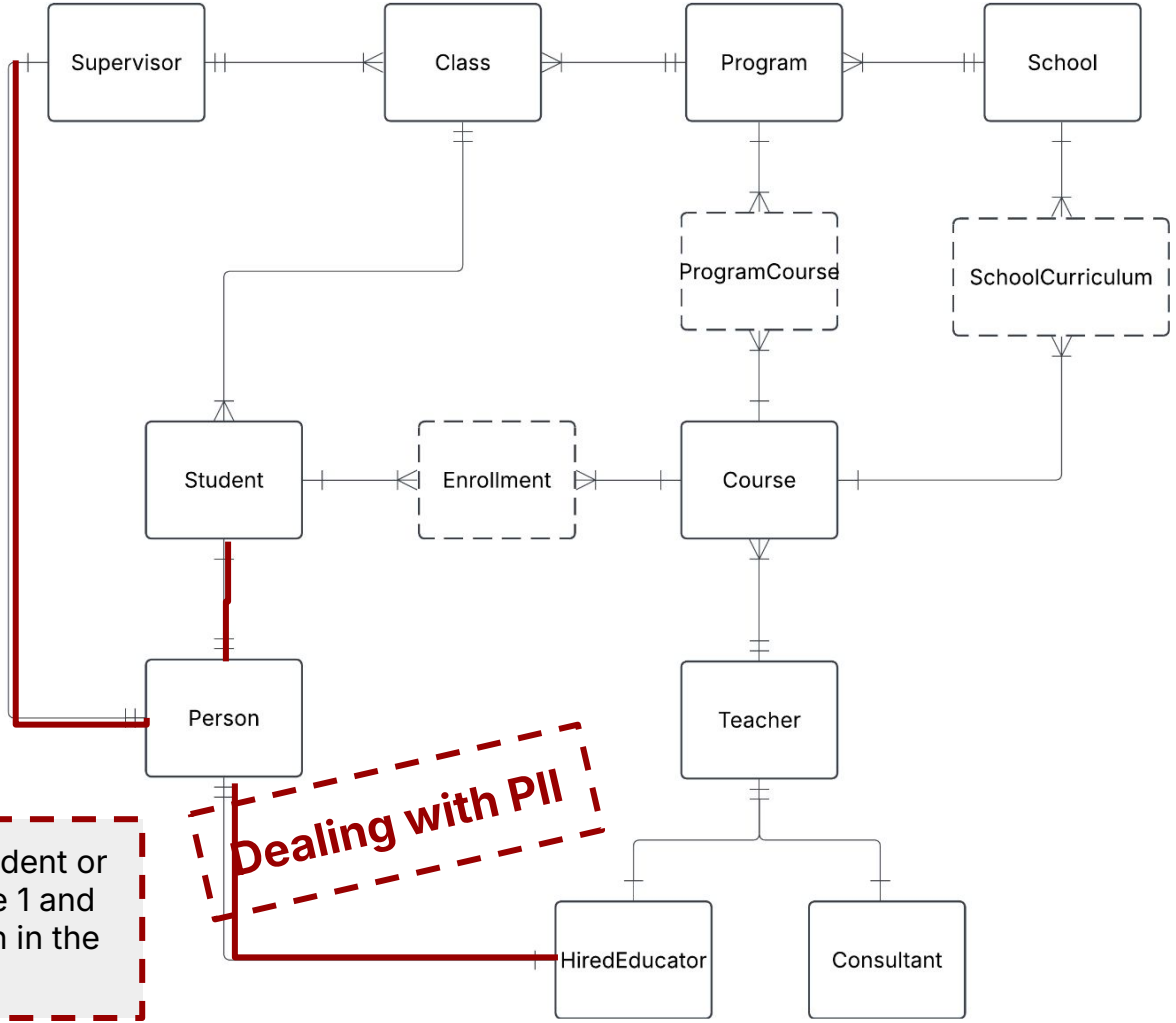
junction table

Normal entity

A Person can have 0 or 1 representation in the HiredEducator, Supervisor and Student table

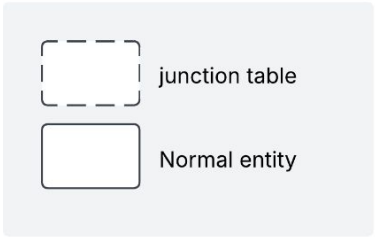
Conceptual model

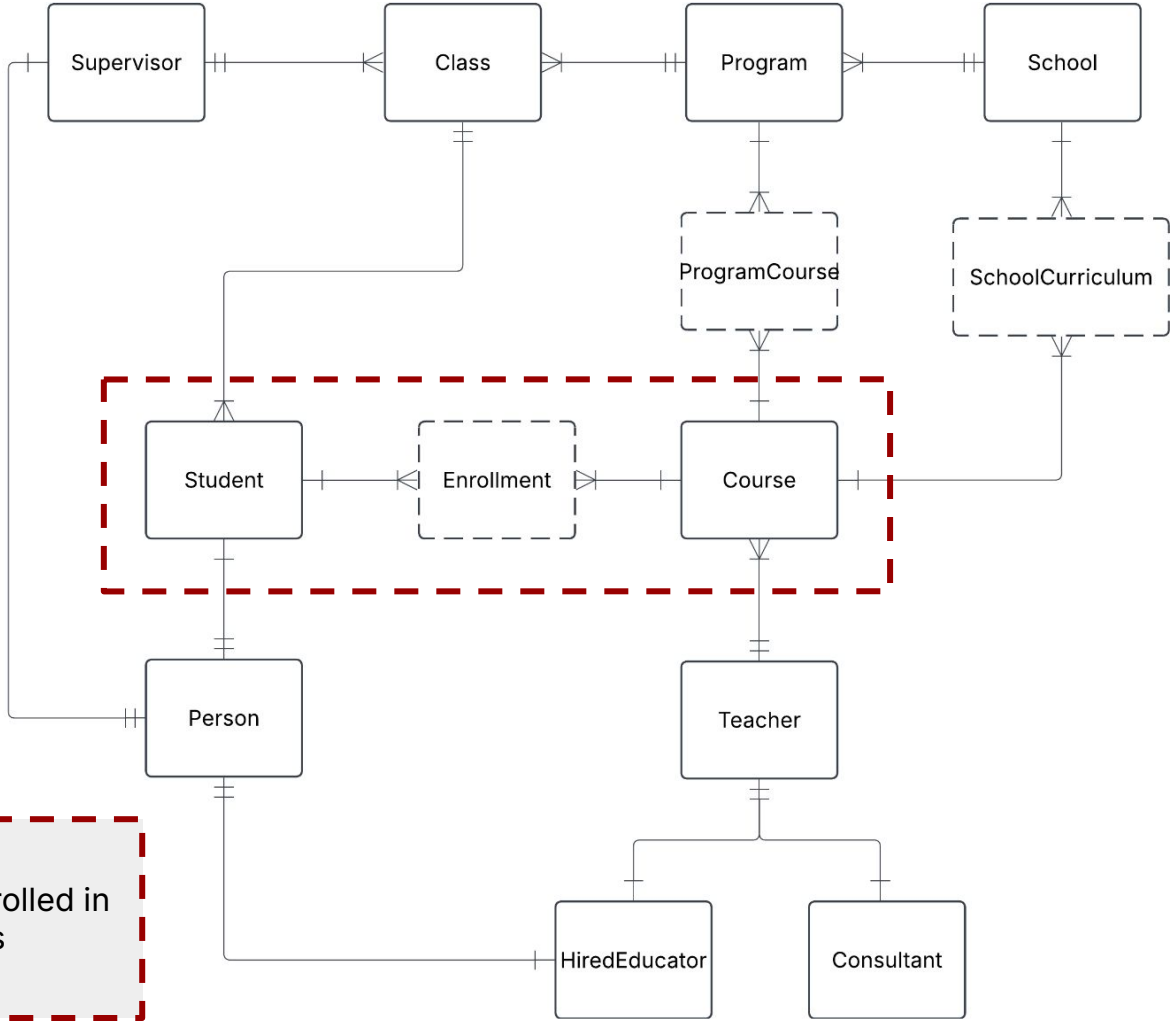




A HiredEducator, Student or Supervisor can have 1 and only 1 representation in the Person table

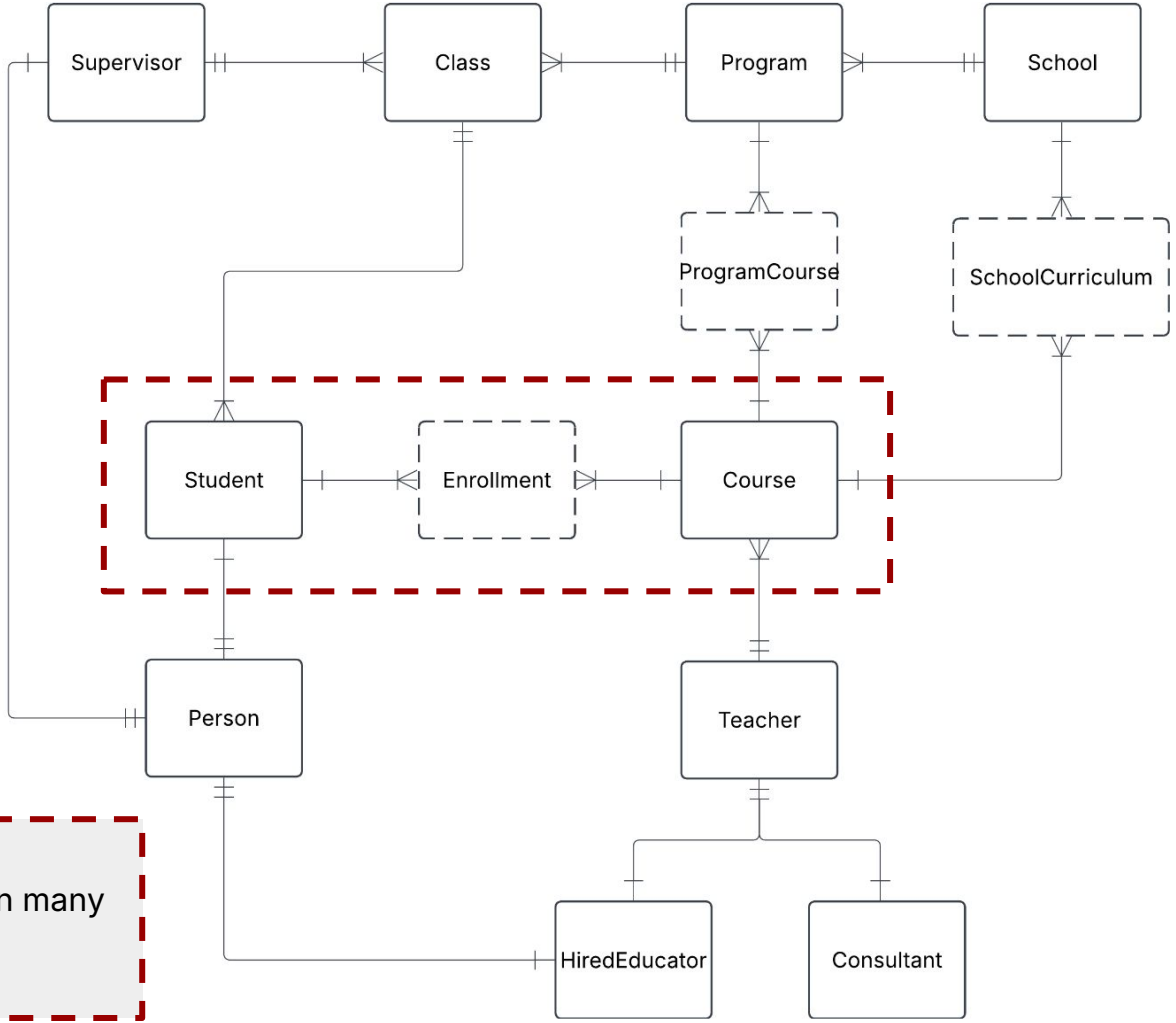
Conceptual model





A Student can be enrolled in many Courses

Conceptual model

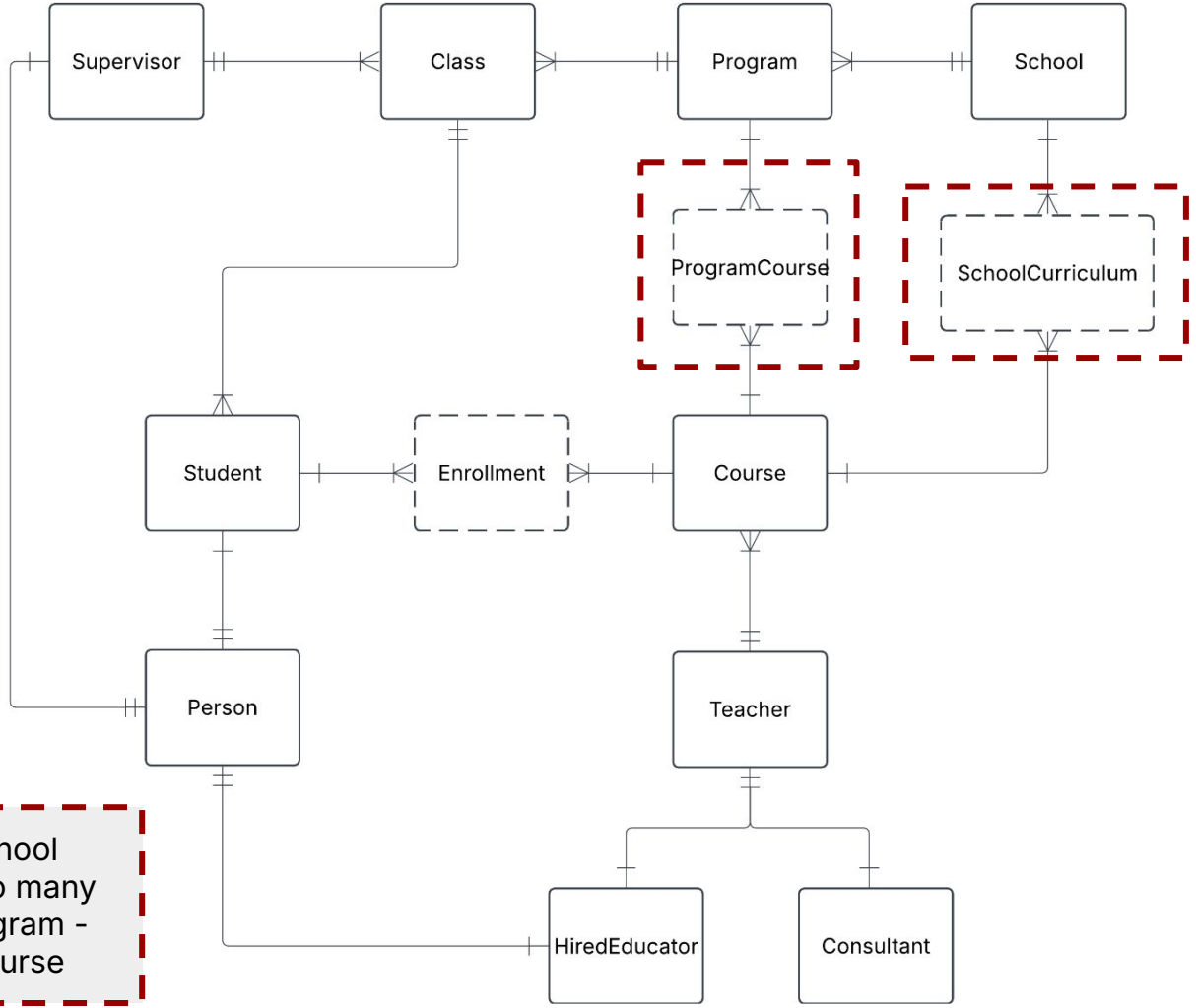


junction table

Normal entity

A Course can contain many Students

Conceptual model

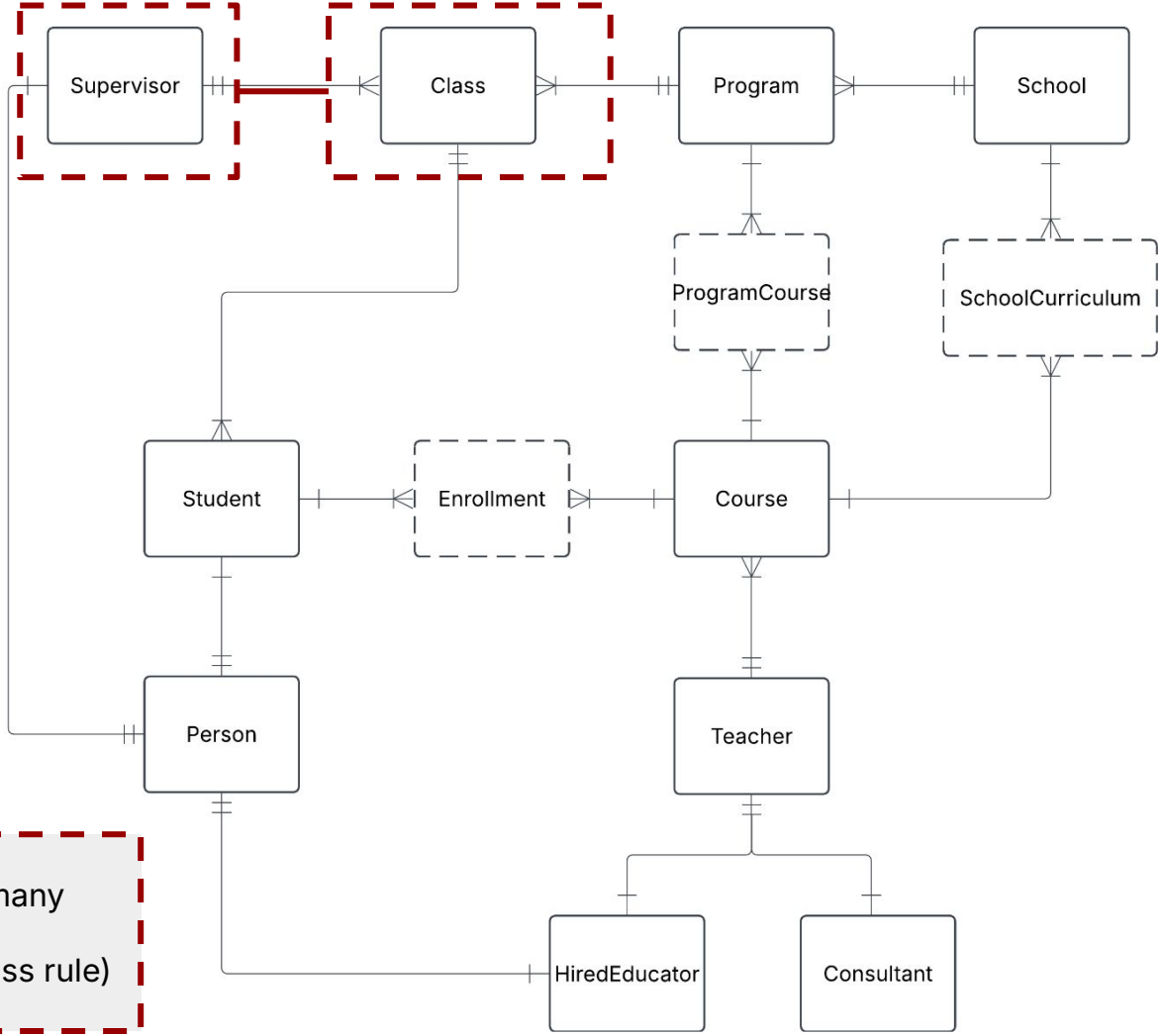


junction table

Normal entity

ProgramCourse and School Curriculum solves many to many relationship between Program - Course and School - Course

Conceptual model

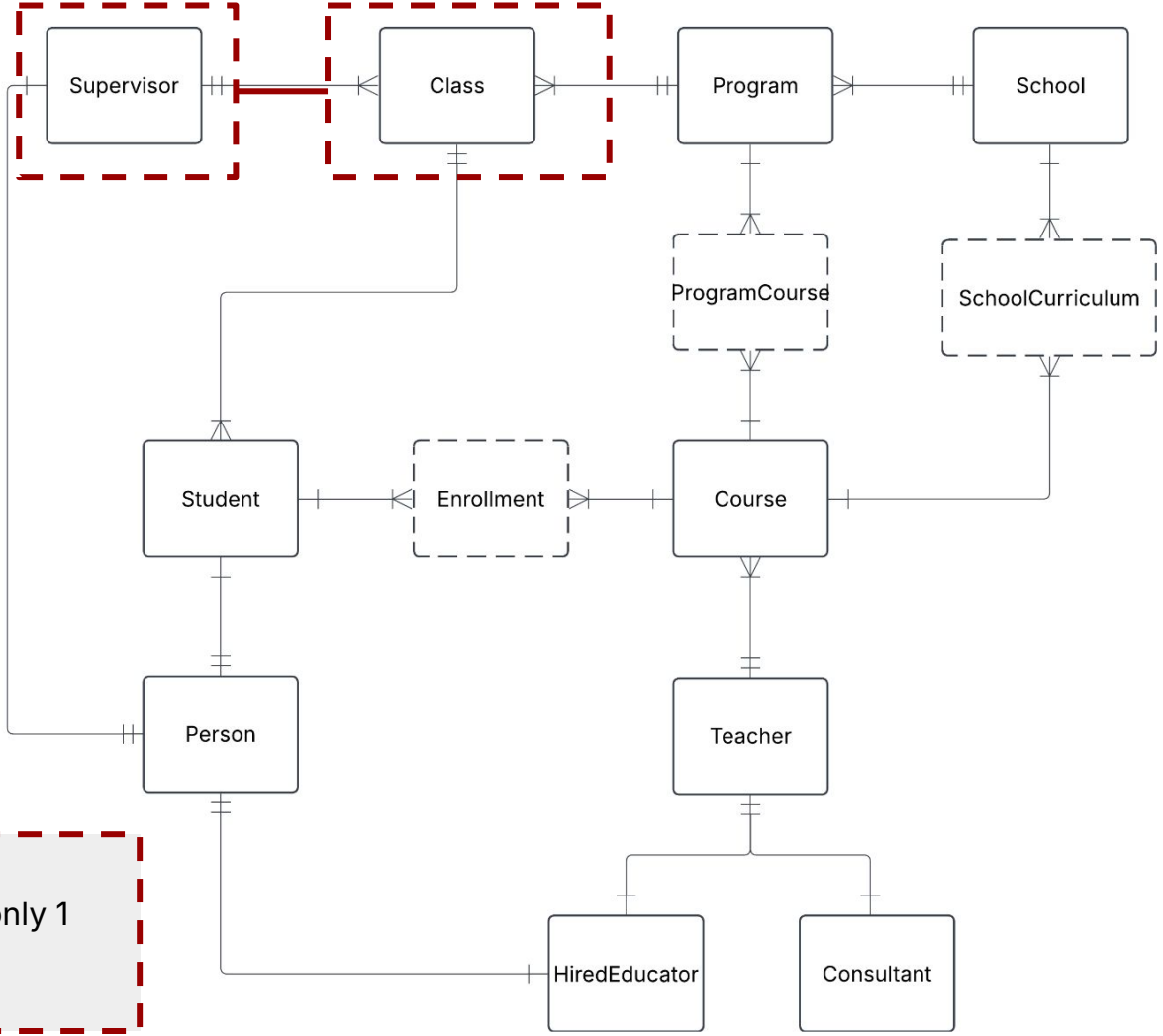


junction table

Normal entity

A Supervisor can have many classes
(can have three as a business rule)

Conceptual model

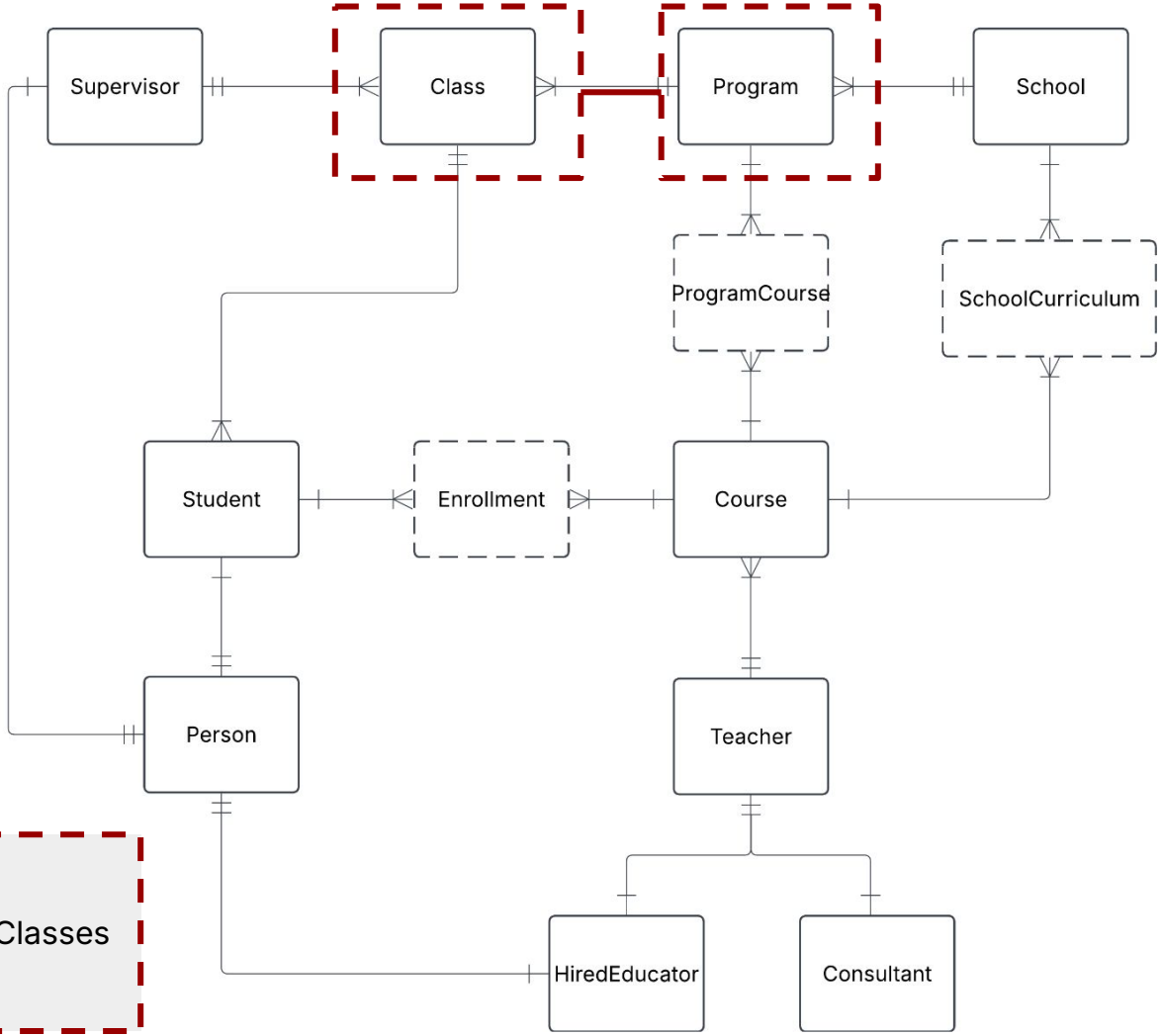


junction table

Normal entity

A Class can have 1 and only 1 Supervisor

Conceptual model

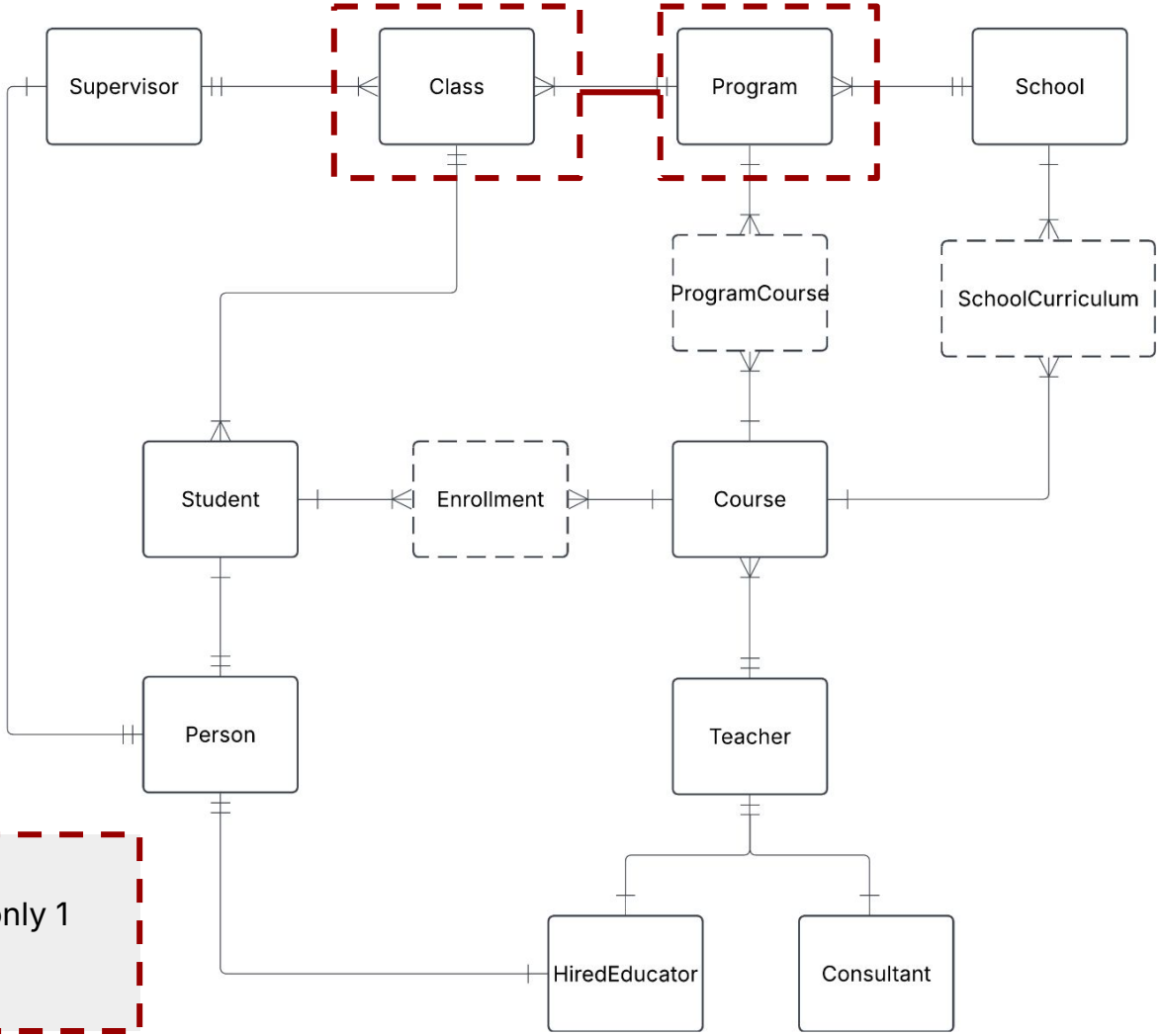


junction table

Normal entity

A Program can have many Classes

Conceptual model

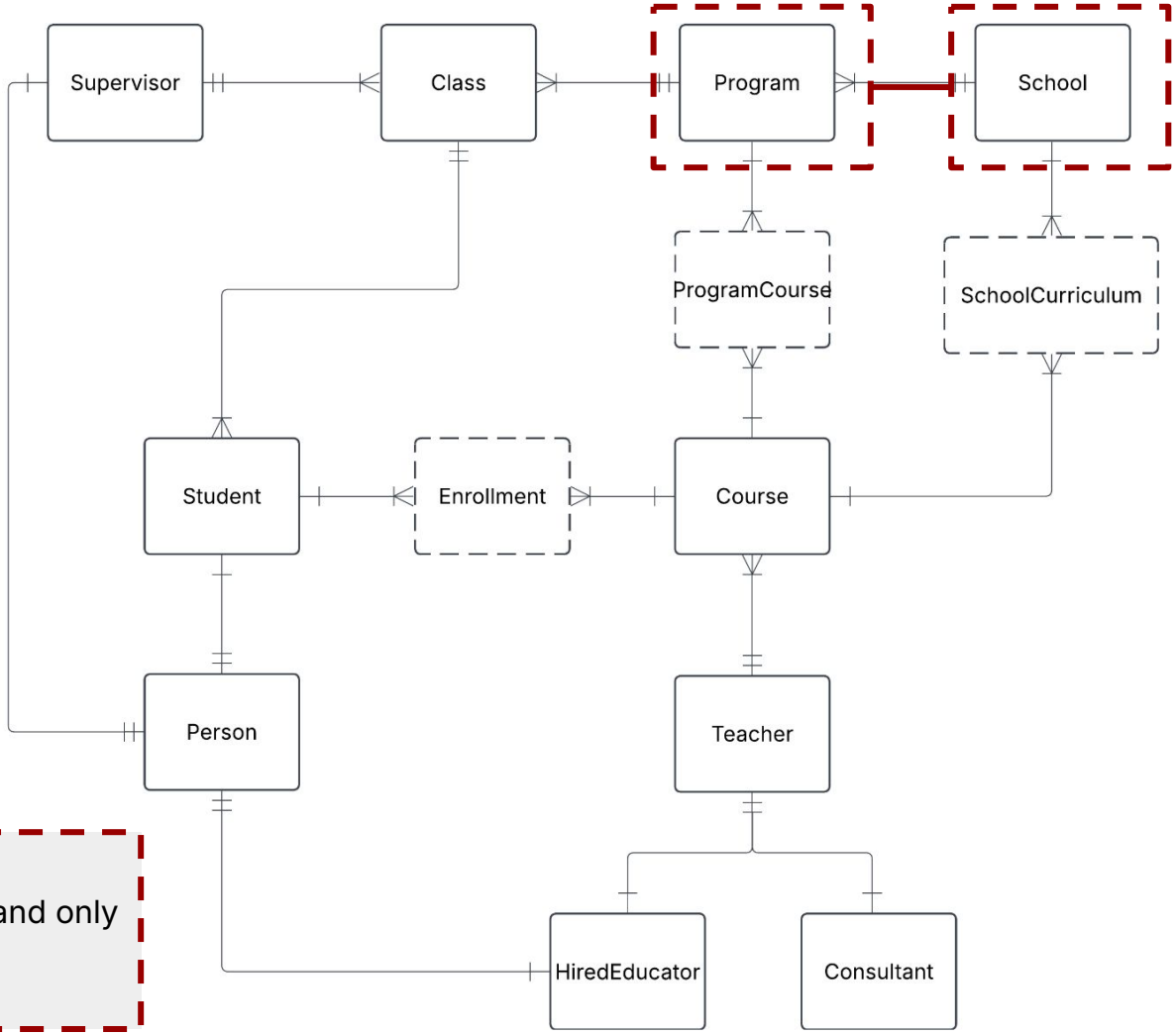


junction table

Normal entity

A Class can have 1 and only 1 Program

Conceptual model

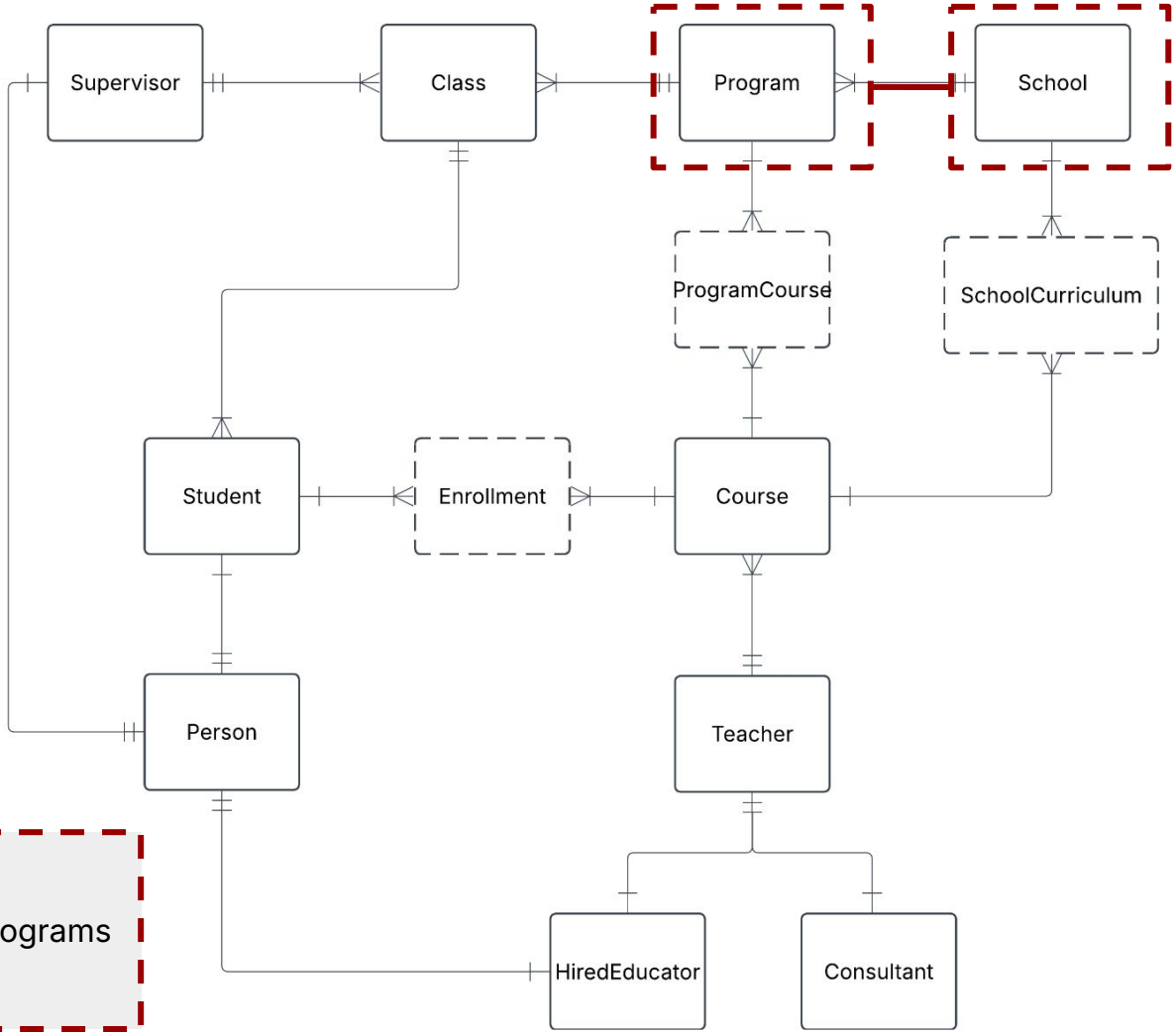


junction table

Normal entity

A Program can be held in 1 and only 1 School

Conceptual model

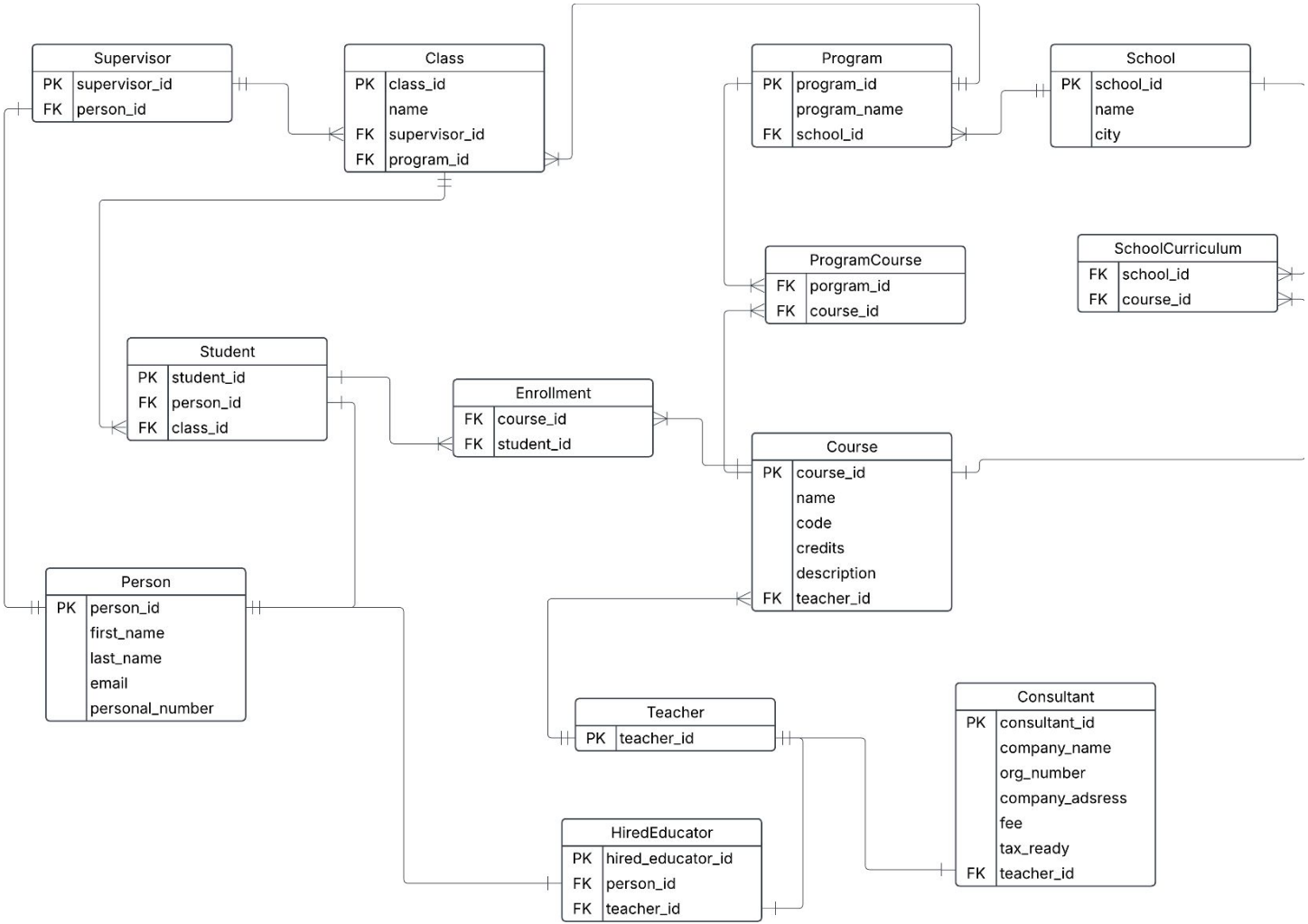


junction table

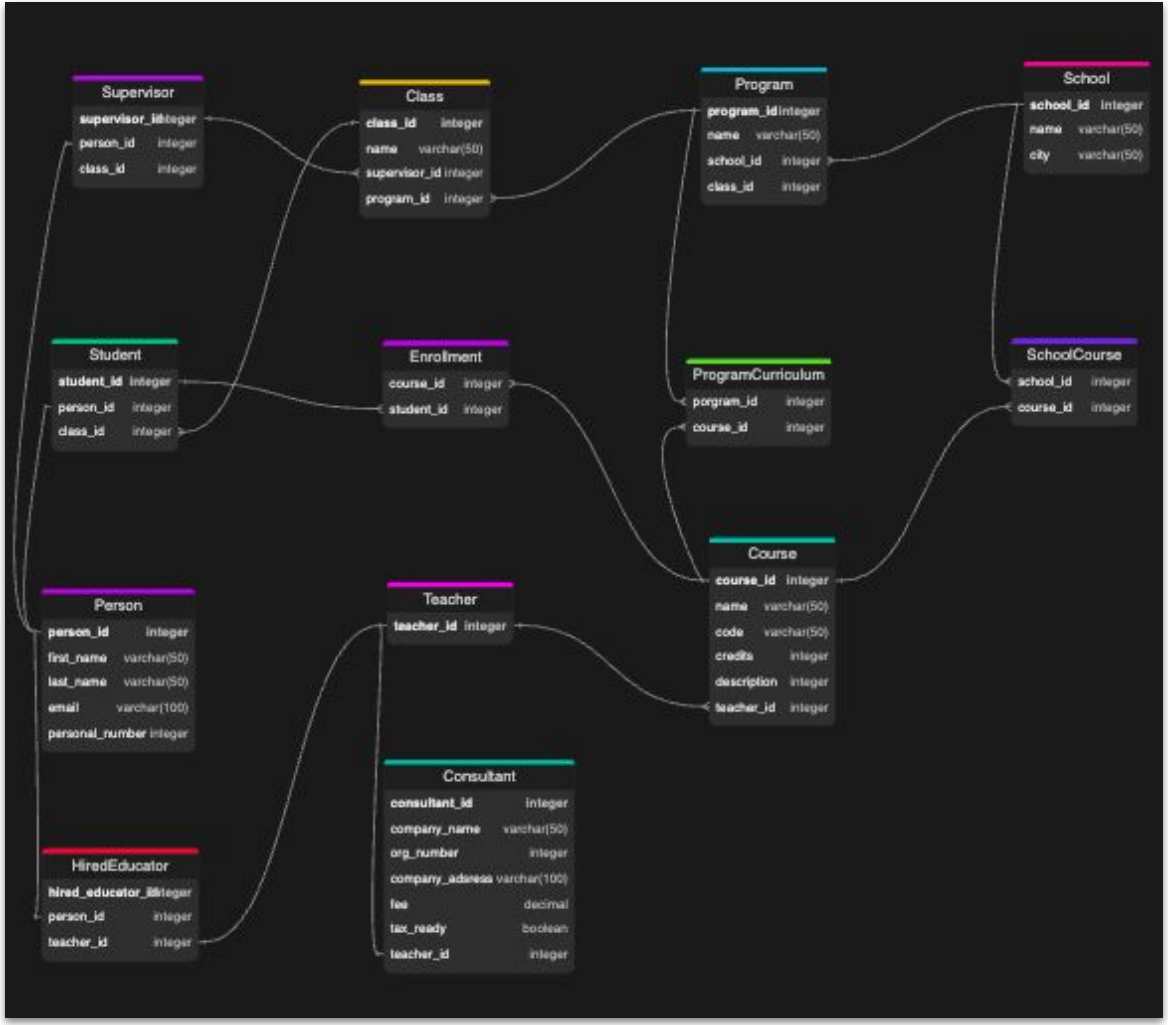
Normal entity

A School can have many Programs

Logical model



Physical model



Physical model

```
TABLE Supervisor {
 supervisor_id integer [PRIMARY KEY]
 person_id integer [ref: - Person.person_id]
 class_id integer
}

TABLE School {
 school_id integer [PRIMARY KEY]
 name varchar(50)
 city varchar(50)
}

TABLE Program {
 program_id integer [PRIMARY KEY]
 name varchar(50)
 school_id integer [ref: > School.school_id]
 class_id integer
}

TABLE Class {
 class_id integer [PRIMARY KEY]
 name varchar(50)
 supervisor_id integer [ref: > Supervisor.supervisor_id]
 program_id integer [ref: > Program.program_id]
}
```

The data

- Dummy data generated in LLM ingested with the purpose to test the relationships and joins
- Methodology for data ingestion

Prompt:
I am modelling a database for a school. See below my sql statement for creating my tables and its cardinalities. I want you to create dummy data in csv's for each of these tables? The dummy data should have the theme Data and AI and courses could be things like AI, Analytics and other development courses. The course names should have a fun twist to the course names. The School's name is YrkesCo and have one location in Stockholm, called "YrkesCo Liljeholmen" and one location in Gothenburg, called "YrkesCo Lindholmen". The student and teachers name should be Swedish sounding names, the first letter in the first_name should be the same as the first letter in the last_name.

The data

- Dummy data generated in LLM ingested with the purpose to test the relationships and joins
- Methodology for data ingestion

Methodology

- Used Gemini to generate dummy data in csv's for each Table
- Added the csv's to my local machine
- Added the csv's with "docker cp" into the container tmp folder in Docker

```
Example Class csv:
docker cp dummy_data/Class.csv postgres_data_modeling:/tmp/Class.csv
```

- Copied the data from the csv's to the database using the COPY command

```
Example Enrollment csv
COPY Enrollment FROM '/tmp/Class.csv' WITH (FORMAT CSV, HEADER);
```

Normalisation - why do we care about it?

- Reduce data redundancy and improve data integrity
  - Changes can be done in place and integrity is ensured
- Prevent anomalies when updating, inserting
  - Ensures high data quality over the entire lifecycle
- Normalisation is a not a silver bullet for OLAP and Data Warehousing
  - For analytics purposes in OLAP systems and data warehousing, denormalization is sometimes desired to optimize read performance for analytical queries for quicker data retrieval.

Normalisation - how it is achieved?

Normal Form	Requirments	Argument
1NF	✅ all tables have primary key ✅ No repeating groups ✅ Uniform column data ✅ Row order does not matter	*Going through each table, all of them has a primary key, the junction tables has it by combining its foreign keys. Each attribute is of one data type and includes no groupings. Row order does matter. Thus, it adhere to first normal form*
2NF	✅ 1NF ✅ Non-prime attributes must be functionally dependendent on entire primary key and not just part of it	*Each table is 1NF. No attributes is functionaly determined by other than the primary key. Thus, 2NF is reached*
3NF	✅ 2NF ✅ Non-prime attributes depends on the key, the whole key and nothing but the key.	*Each table is 2NF. There are no transitive dependencies of the attributes. Thus, 3NF is reached*

Some query results and why they matter to the business

Query: for each school, list the program names and the courses offered within those programs?

Why it matters: This is showing that the school sites can are having distinct programs as modelled. It also shows that more school sites can be added in the future if the business grows.

	school_name character varying	program_name character varying	course_name character varying	code character varying	description character varying
1	YrkesCo Liljeholmen	Data Science Mastery	Pythonic Data Delights	PYDD101	Python for data analysis
2	YrkesCo Liljeholmen	Data Science Mastery	Neural Network Nirvana	NNN201	Deep learning foundations
3	YrkesCo Liljeholmen	AI Innovation Lab	Neural Network Nirvana	NNN201	Deep learning foundations
4	YrkesCo Liljeholmen	AI Innovation Lab	AI & Algorithm Alchemy	AAAA501	Advanced AI and algorithms
5	YrkesCo Lindholmen	Full Stack Web Dev	React & Roll	RR301	Web development with React
6	YrkesCo Lindholmen	Data Analytics Pro	Data Dive & Discover	DDD401	Data visualization & exploration
7	YrkesCo Liljeholmen	Machine Learning Wizardry	AI & Algorithm Alchemy	AAAA501	Advanced AI and algorithms
8	YrkesCo Lindholmen	Cloud Engineering	Cloud Computing Chronicles	CCC601	Cloud computing essentials
9	YrkesCo Liljeholmen	Data Science Mastery	Database Design Dazzle	DDDD701	Database design and SQL

Some query results and why they matter to the business

Query: what courses belong to more than one program?

Why it matters: This is showing that the school can have shared courses across programs as defined in the business requirements.

	course_name character varying	nr_courses bigint
1	AI & Algorithm Alchemy	2
2	Neural Network Nirvana	2

Some query results and why they matter to the business

Query: For each program, list the names of the students and their personal number enrolled in the classes within that program.

Why it matters: This is showing that simplicity of joining tables to get lists of personal information per course

	program_name character varying	course_name character varying	first_name character varying	last_name character varying	personal_number character varying
1	Data Science Mastery	Pythonic Data Delights	Niklas	Nilsson	198402204567
2	Data Science Mastery	Pythonic Data Delights	Greta	Gustafsson	199107156789
3	Data Science Mastery	Pythonic Data Delights	Fredrik	Forsberg	198306102345
4	Data Science Mastery	Neural Network Nirvana	Olivia	Olsson	199703258901
5	Data Science Mastery	Neural Network Nirvana	Ida	Isaksson	199409254567
6	Data Science Mastery	Neural Network Nirvana	Hanna	Holm	198708200123
7	AI Innovation Lab	Neural Network Nirvana	Olivia	Olsson	199703258901
8	AI Innovation Lab	Neural Network Nirvana	Ida	Isaksson	199409254567
9	AI Innovation Lab	Neural Network Nirvana	Hanna	Holm	198708200123
10	AI Innovation Lab	AI & Algorithm Alchemy	Fredrik	Forsberg	198306102345
11	AI Innovation Lab	AI & Algorithm Alchemy	Olivia	Olsson	199703258901
12	AI Innovation Lab	AI & Algorithm Alchemy	Niklas	Nilsson	198402204567
13	Full Stack Web Dev	React & Roll	Per	Persson	198104302345
14	Full Stack Web Dev	React & Roll	Karin	Karlsson	199311052345
15	Full Stack Web Dev	React & Roll	Johan	Jonsson	198210308901
16	Data Analytics Pro	Data Dive & Discover	Quirine	Qvist	199805056789
17	Data Analytics Pro	Data Dive & Discover	Maria	Magnusson	199601150123
18	Data Analytics Pro	Data Dive & Discover	Lars	Lindberg	198612106789
19	Machine Learning Wizardry	AI & Algorithm Alchemy	Fredrik	Forsberg	198306102345
20	Machine Learning Wizardry	AI & Algorithm Alchemy	Olivia	Olsson	199703258901
21	Machine Learning Wizardry	AI & Algorithm Alchemy	Niklas	Nilsson	198402204567
22	Cloud Engineering	Cloud Computing Chronicles	Greta	Gustafsson	199107156789
23	Cloud Engineering	Cloud Computing Chronicles	Quirine	Qvist	199805056789
24	Cloud Engineering	Cloud Computing Chronicles	Per	Persson	198104302345
25	Data Science Mastery	Database Design Dazzle	Hanna	Holm	198708200123

Thank you for listening...