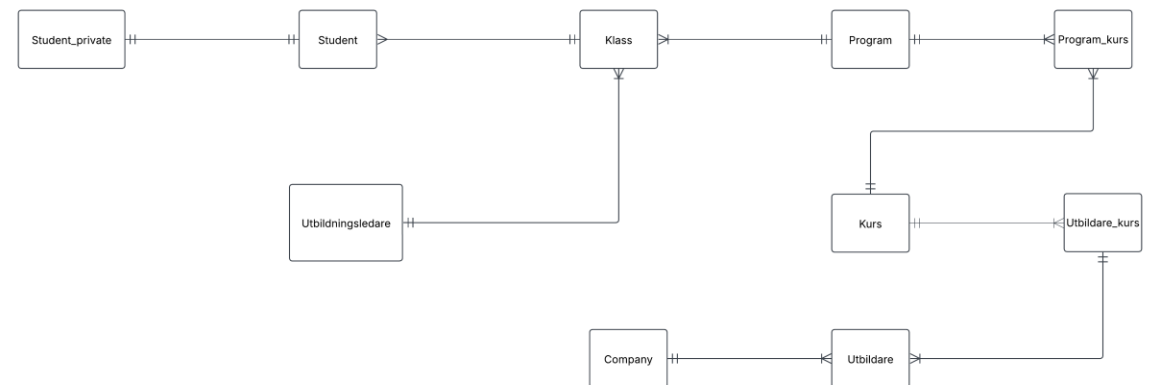


Datamodellering av YH-utbildning

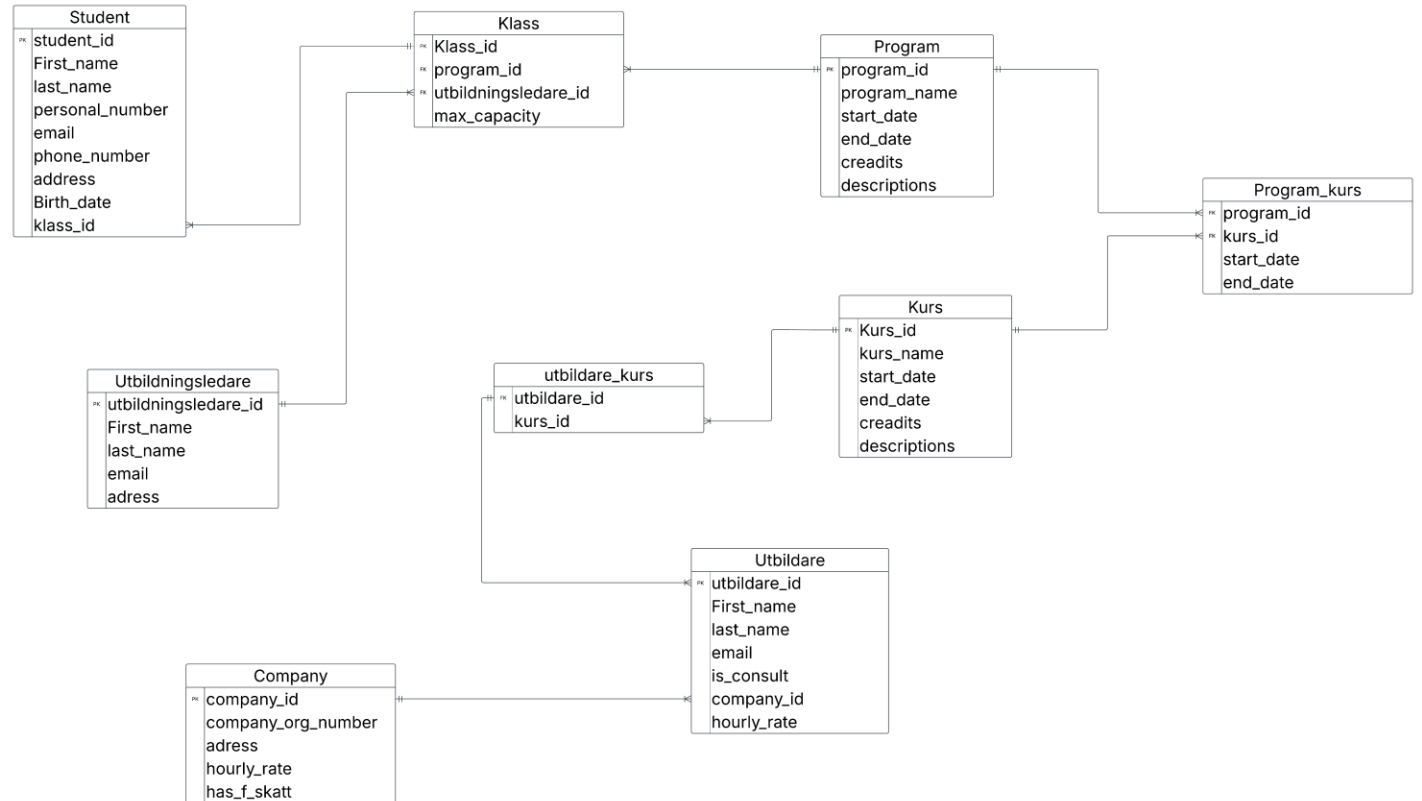
ER-modell, databasstruktur
och implementation

Abdulazeez Stef

DE24

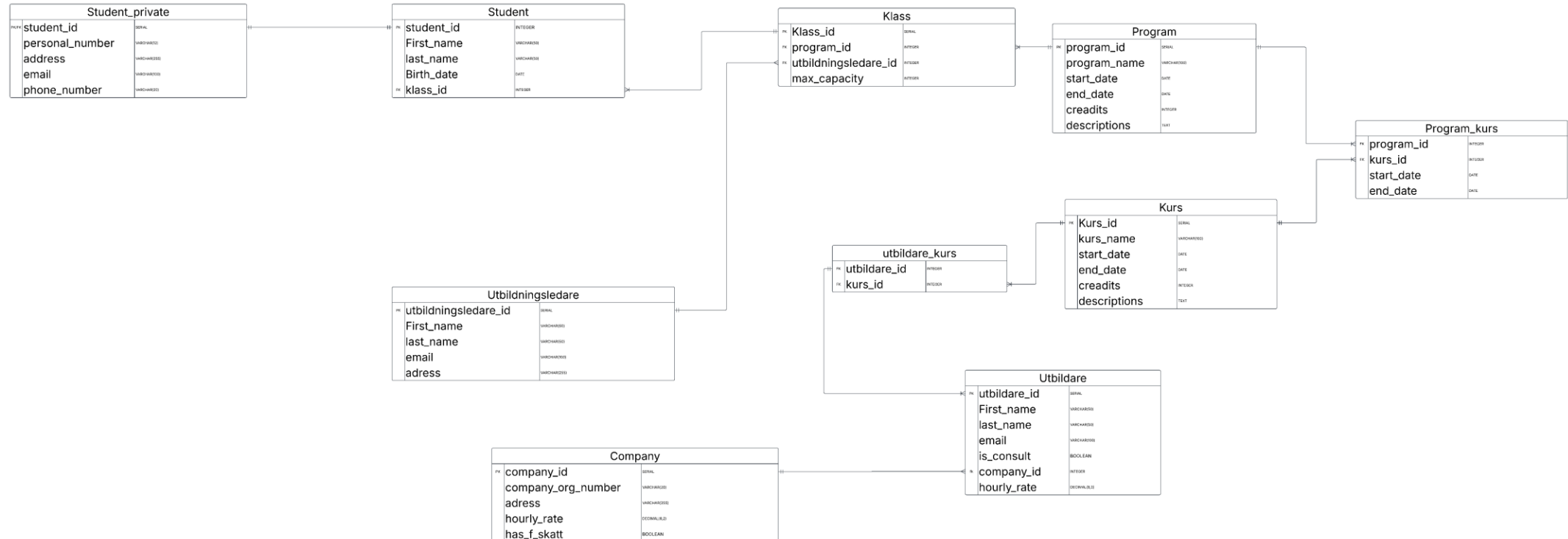


Syfte med datamodellen



- Skapa en databas för en YH-utbildning
- Hantera studenter, kurser, program, utbildare och företag
- Möjlighet till framtida utbyggnad
- Tydlig struktur för relationer

ER-modellen



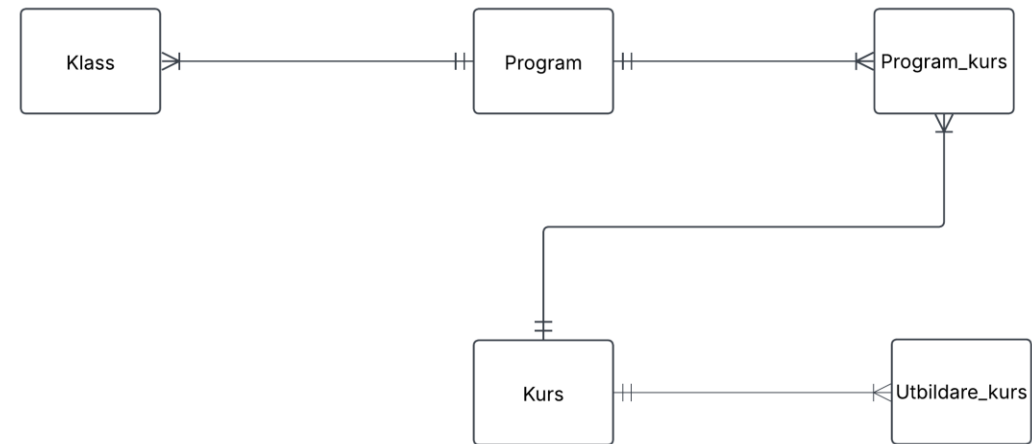
Student & Student_private

- Student innehåller allmän information
- Student_private innehåller känsliga uppgifter
- Kopplade via student_id



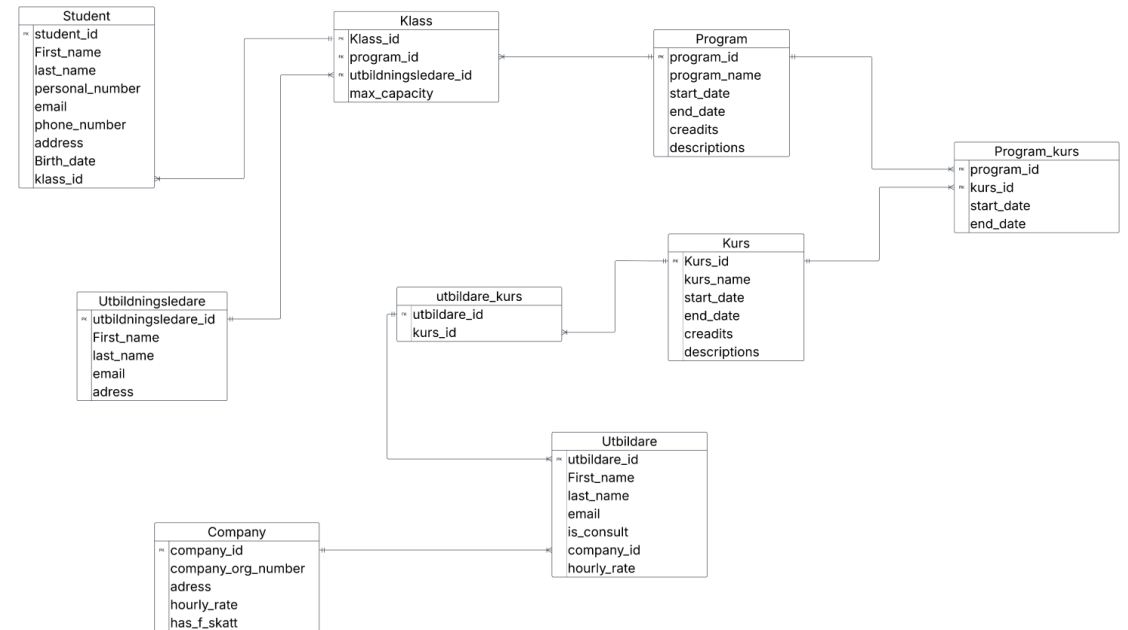
Klass, Program & Kurs

- En klass tillhör ett program
- Ett program består av flera kurser
- Kurserna har start- och slutdatum



Utbildningsledare & Utbildare

- Varje klass har en utbildningsledare
- Kurser undervisas av utbildare
- Utbildare kan vara anställda eller konsulter



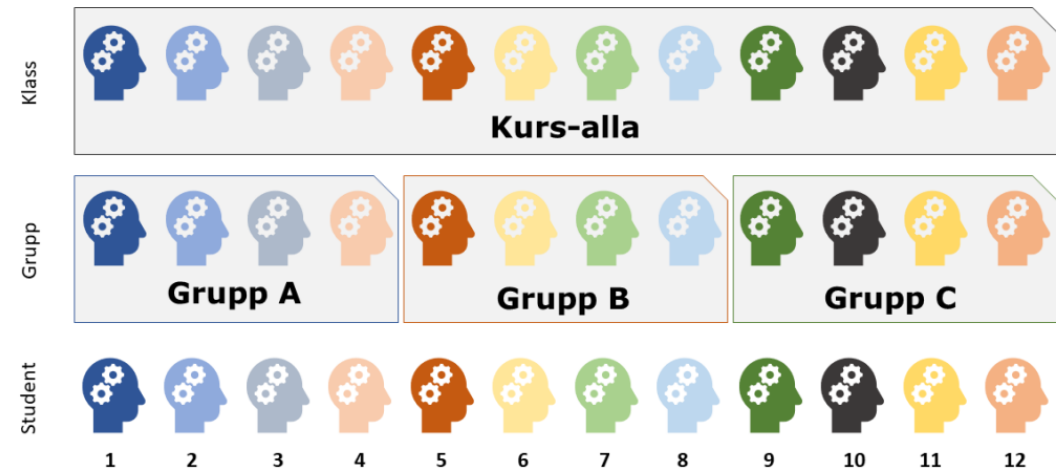
Företag för konsulter

- Företag har organisationsnummer, adress, F-skatt
- Kopplas till utbildare vid behov
- Timpris sparas i både Company och Utbildare



Exempel på relationer

- En student tillhör en klass
- En klass har ett program
- Ett program består av kurser
- Kurser har utbildare



Implementation i SQL

SERIAL, VARCHAR, DATE,
PRIMARY KEY, FOREIGN KEY,
UNIQUE

```
--Student_private
CREATE TABLE Student_private (
    student_id SERIAL PRIMARY KEY,
    personal_number VARCHAR(12),
    address VARCHAR(255),
    email VARCHAR(100)
);

--Student
CREATE TABLE Student (
    student_id INTEGER PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    phone_number VARCHAR(20),
    birth_date DATE,
    klass_id INTEGER,
    FOREIGN KEY (student_id) REFERENCES Student_private(student_id),
    FOREIGN KEY (klass_id) REFERENCES Klass(klass_id)
);
```

Testdata & SQL-frågor

```
insert_data.sql
1  -- 1. Program (minst 1 program)
2  INSERT INTO Program (program_name, start_date, end_date, credits, description) VALUES
3  ('Systemutvecklare .NET', '2024-08-20', '2026-06-15', 400, 'Tvåårigt YH-program');
4
5  -- 2. Utbildningsledare (minst 1)
6  INSERT INTO Utbildningsledare (first_name, last_name, email, address) VALUES
7  ('Anna', 'Andersson', 'anna.andersson@yh.se', 'Stockholm');
8
9  -- 3. Klasser (1 klass per program)
10 INSERT INTO Klass (program_id, utbildningsledare_id, max_capacity) VALUES
11 (1, 1, 30); -- En klass för .NET-programmet
12
13 -- 4. Studenter (minst 3 studenter)
14 INSERT INTO Student (first_name, last_name, personal_number, email, klass_id) VALUES
15 ('Erik', 'Svensson', '199001011234', 'erik.svensson@student.se', 1),
16 ('Maria', 'Karlsson', '199502028765', 'maria.karlsson@student.se', 1),
17 ('Johanna', 'Pettersson', '199604151234', 'johanna.pettersson@student.se', 1);
18
19 -- 5. Kurser (minst 1 kurs)
20 INSERT INTO Kurs (kurs_name, credits, description) VALUES
21 ('Databasteknik', 20, 'Grundläggande SQL');
22
23 -- 6. Program_kurs (koppla kurs till program)
24 INSERT INTO Program_kurs (program_id, kurs_id) VALUES
25 (1, 1); -- Databasteknik ingår i .NET-programmet
26
27 -- 7. Company (om du har konsulter)
28 INSERT INTO Company (company_org_number, has_f_skatt) VALUES
29 ('556677-8899', TRUE);
30
```

SQL-frågor

```
queries.sql
1  SELECT
2      s.student_id,
3      s.first_name,
4      s.last_name,
5      u.first_name AS ledare_fornamn,
6      u.last_name AS ledare_efternamn
7  FROM
8      Student s
9  JOIN
10     Klass k ON s.klass_id = k.klass_id
11  JOIN
12     Utbildningsledare u ON k.utbildningsledare_id = u.utbildningsledare_id
13  WHERE
14     k.klass_id = 1; -- Exempel: klass_id 1
15
16
17  SELECT
18     p.program_name,
19     k.kurs_name,
20     k.creddfis AS credits,
21     ub.first_name AS utbildare_fornamn,
22     ub.last_name AS utbildare_efternamn,
23     CASE
24         WHEN ub.is_consult = TRUE THEN 'Konsult'
25         ELSE 'Fast anställd'
26     END AS anställningstyp
27  FROM
28     Program p
29  JOIN
```

Slutsats och reflektion

Tydlig struktur med normalisering

Bra för framtida utveckling

Separat hantering av känslig data



THANK
YOU