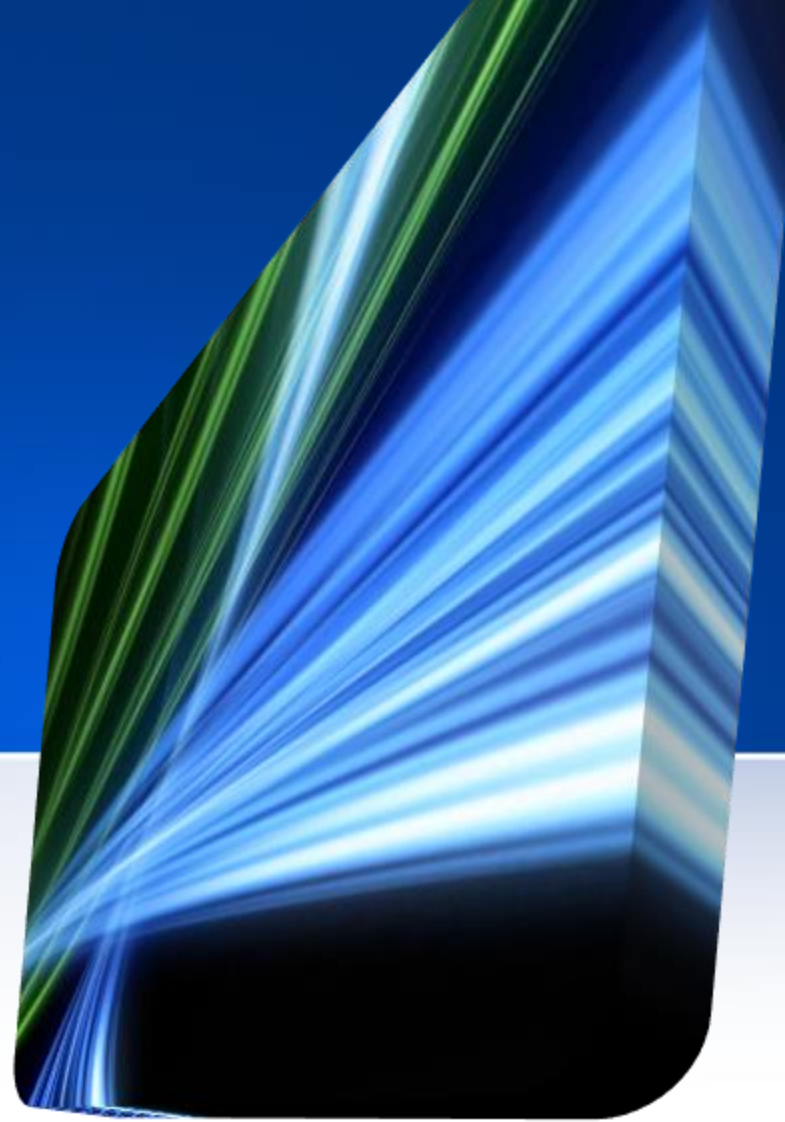


# YrkesCO projekt

- **YrkesCo – Databasprojekt**
- 📁 **Titel:**  
*Databassystem för utbildningsorganisationen YrkesCo*
- 👤 Mustafa Hassan Mahamud
- 🎓 **Kurs:**  
Datamodeling/ Data engineer
- 🏰 Stockholm tekniska
- 📅 **Datum:**  
6 maj 2025



# Introduktion



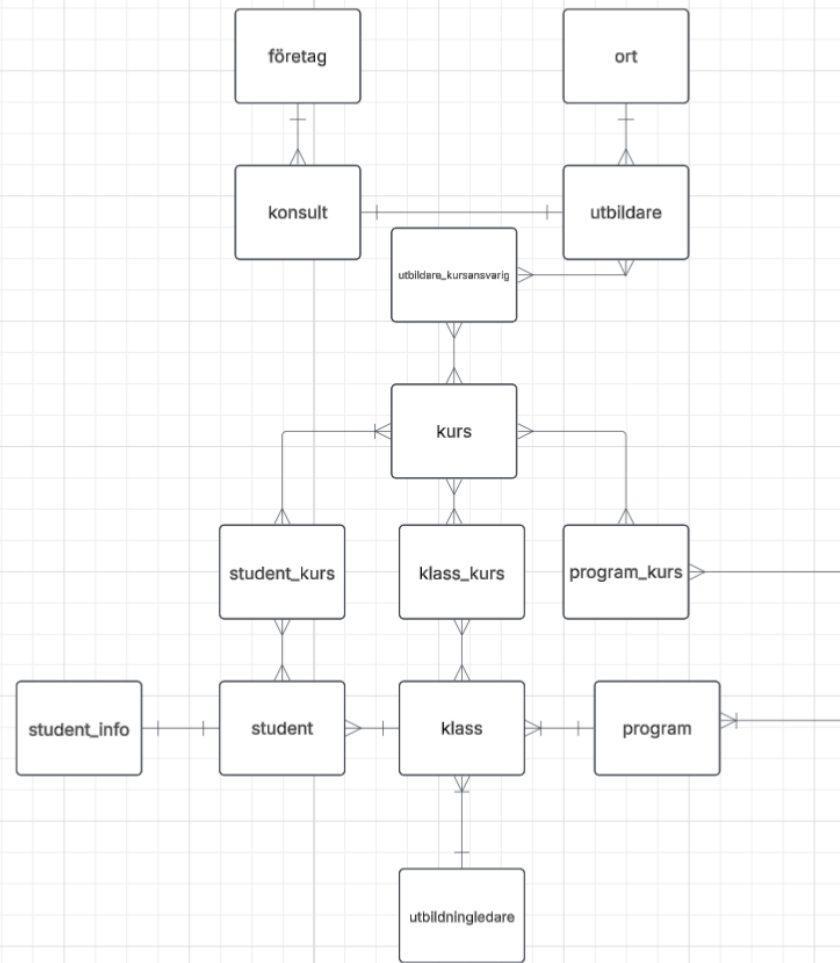
- Syfte:  
Syftet med uppgiften är att designa och implementera en relationsdatabas för ett utbildningssystem. Fokus ligger på att skapa en välstrukturerad datamodell med tydliga relationer mellan tabeller som representerar orter, utbildare, studenter, kurser och program. Genom att använda SQL skapas och fylls databasen med data, samt körs frågor för att hämta relevant information.

## 2. Databasschema (ER-diagram & fysisk modell)

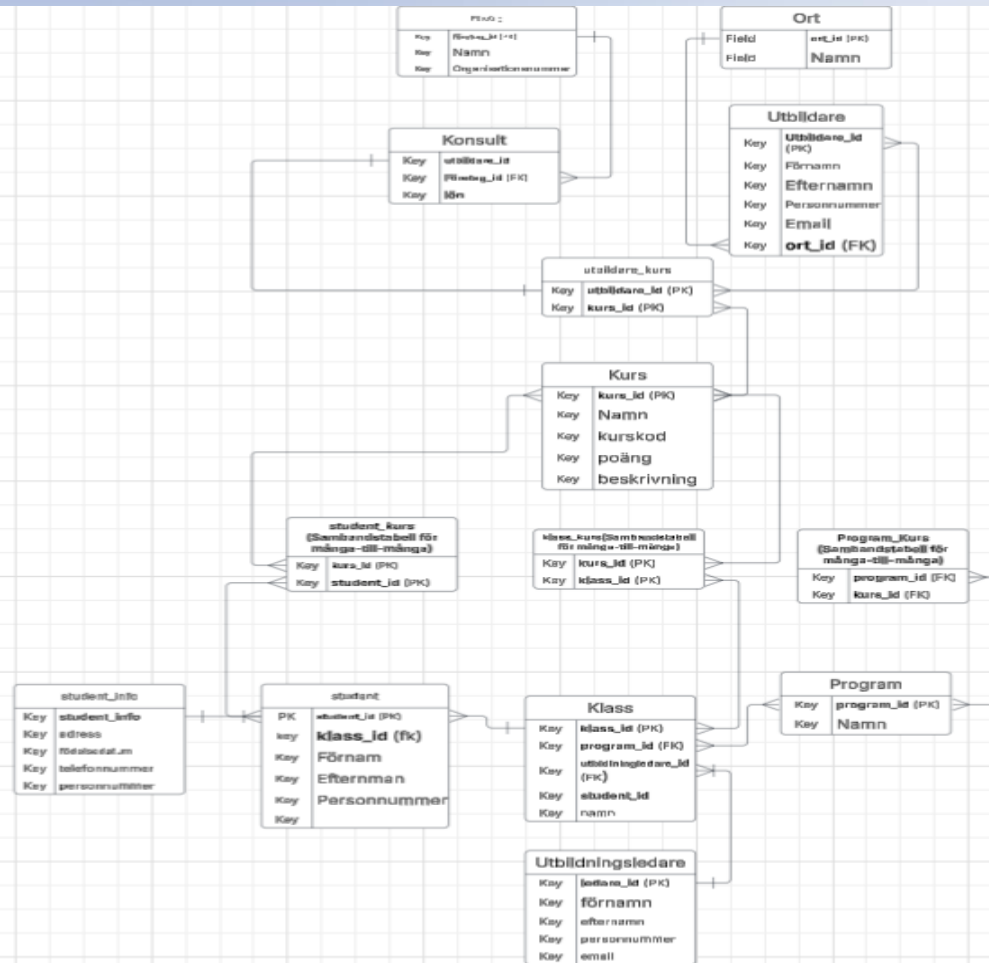


- konceptuell modell
- Logisk modell
- Fysisk modell

# Konceptuell modell



# Logisk modell



# Fysisk modell



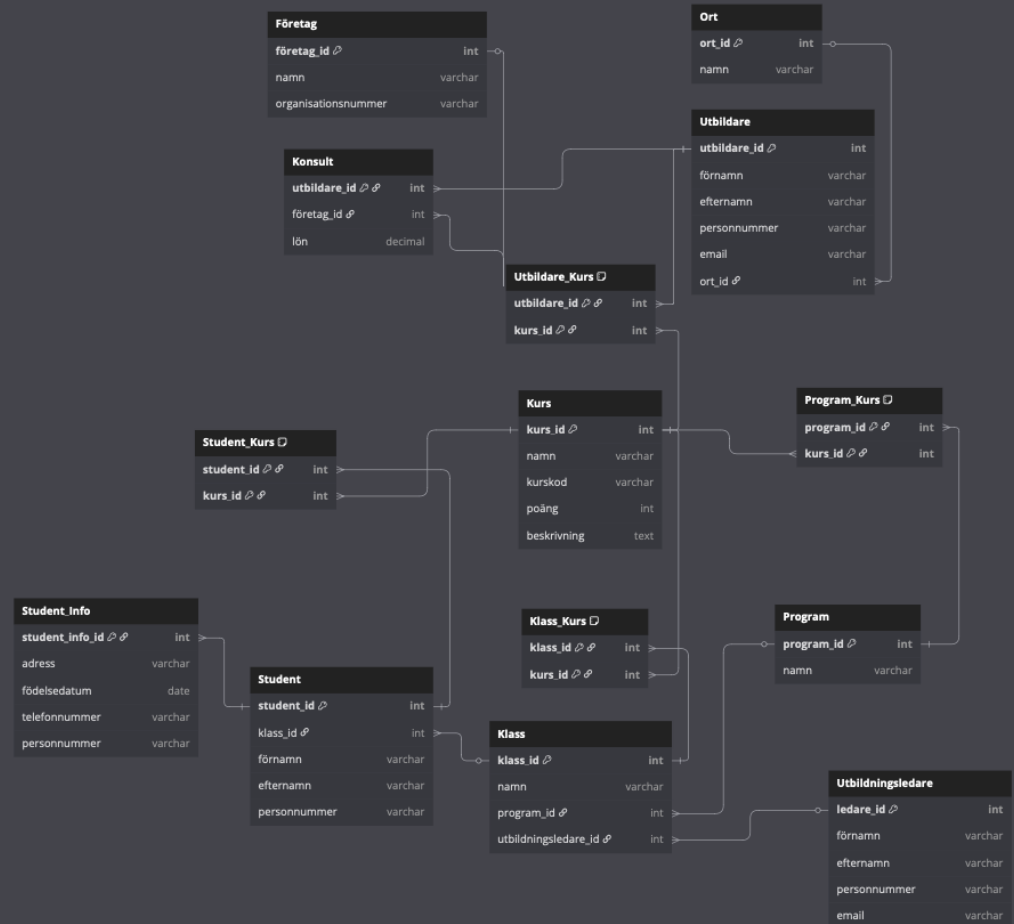
```
// Schema: FKRESU
```

```
Table Ort {  
  ort_id int [pk, increment]  
  namn varchar  
}
```

```
Table Företag {  
  företag_id int [pk, increment]  
  namn varchar  
  organisationsnummer varchar  
}
```




```
Table Utbildare {  
  utbildare_id int [pk, increment]  
  förnamn varchar  
  efternamn varchar  
  personnummer varchar  
  email varchar  
  ort_id int [ref: > Ort.ort_id]  
}
```

```
Table Konsult {  
  utbildare_id int [pk, ref: > Utbildare.utbildare_id]  
  företag_id int [ref: > Företag.företag_id]  
  lön decimal  
}
```




# normalisera databasen till tredje normalformen



- **Varför 3NF är viktigt**
-  **Mindre datadubbling** – Varje uppgift lagras bara en gång.
-  **Enklare uppdateringar** – Ändringar görs på ett ställe.
-  **Bättre datakvalitet** – Varje tabell fokuserar på en sak.
- **Undviker datadubbling och sparar utrymme**  
3NF säkerställer att uppgifter som ortnamn och kursbeskrivning lagras på ett ställe. Det gör databasen effektivare och lättare att underhålla.
- **Flexibel och enkel att uppdatera**  
Ändras en ort eller ett organisationsnummer, räcker det att uppdatera en tabell. Det minskar risken för fel.
- **Bättre datakvalitet och tydliga relationer**  
Varje tabell har ett tydligt syfte och beroenden går direkt till primärnyckeln. Exempelvis är Student\_Info separat från Student för bättre struktur.

# Kod & Implementering



-  Testdata
- Postgres
- SQL
- Lucidchart

```
JOIN yrkesco.klass k ON s.klass_id = k.klass_id
JOIN yrkesco.program p ON k.program_id = p.program_id;
förnamn | efternamn | klass | program
-----|-----|-----|-----
Mustafa | Hassan Mahamud | DE23 | Data Engineer
(1 row)

data_modeling_course_db=> SELECT pr.namn AS program, ku.namn AS kurs
FROM yrkesco.program_kurs pk
JOIN yrkesco.program pr ON pk.program_id = pr.program_id
JOIN yrkesco.kurs ku ON pk.kurs_id = ku.kurs_id
ORDER BY pr.namn;
program | kurs
-----|-----
Data Engineer | Databasteknik
Data Engineer | Python-programmering
(2 rows)

data_modeling_course_db=> 44%4
```

```
data_modeling_course_db=> SELECT förnamn, efternamn, email FROM yrkesco.student;
förnamn | efternamn | email
-----|-----|-----
Mustafa | Hassan Mahamud | mustafa@student.se
Ali | Yusuf | ali@student.se
Amina | Mohamed | amina@student.se
Mustafa | Hassan Mahamud | mustafa@student.se
Ali | Yusuf | ali@student.se
Amina | Mohamed | amina@student.se
(6 rows)

data_modeling_course_db=> SELECT s.förnamn, s.etternamn, k.namn AS klass, p.namn AS program
FROM yrkesco.student s
JOIN yrkesco.klass k ON s.klass_id = k.klass_id
JOIN yrkesco.program p ON k.program_id = p.program_id;
förnamn | efternamn | klass | program
-----|-----|-----|-----
Mustafa | Hassan Mahamud | DE23 | Data Engineer
```


```
CREATE SCHEMA IF NOT EXISTS yrkesco;

SET search_path to yrkesco;
```



# Sammafattning



- **Avslutning**
- Vi har skapat en välstrukturerad databas med tydliga relationer mellan studenter, kurser, utbildare och program. Systemet är redo att användas för administration och vidareutveckling.
-  **Frågor?**  
Tack för er uppmärksamhet!