

# Databas för YrkesCo

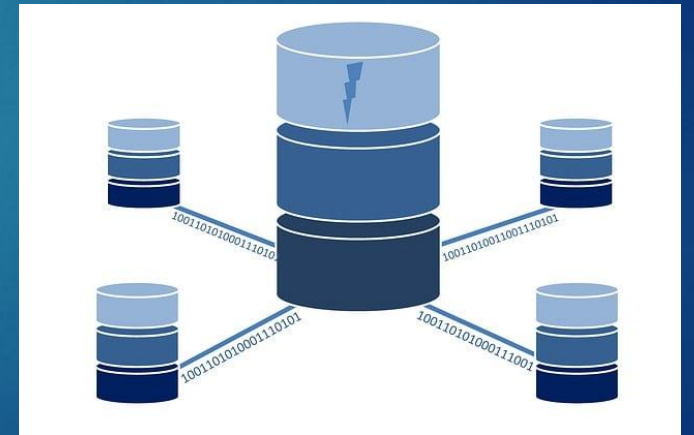


Hej och välkomna!



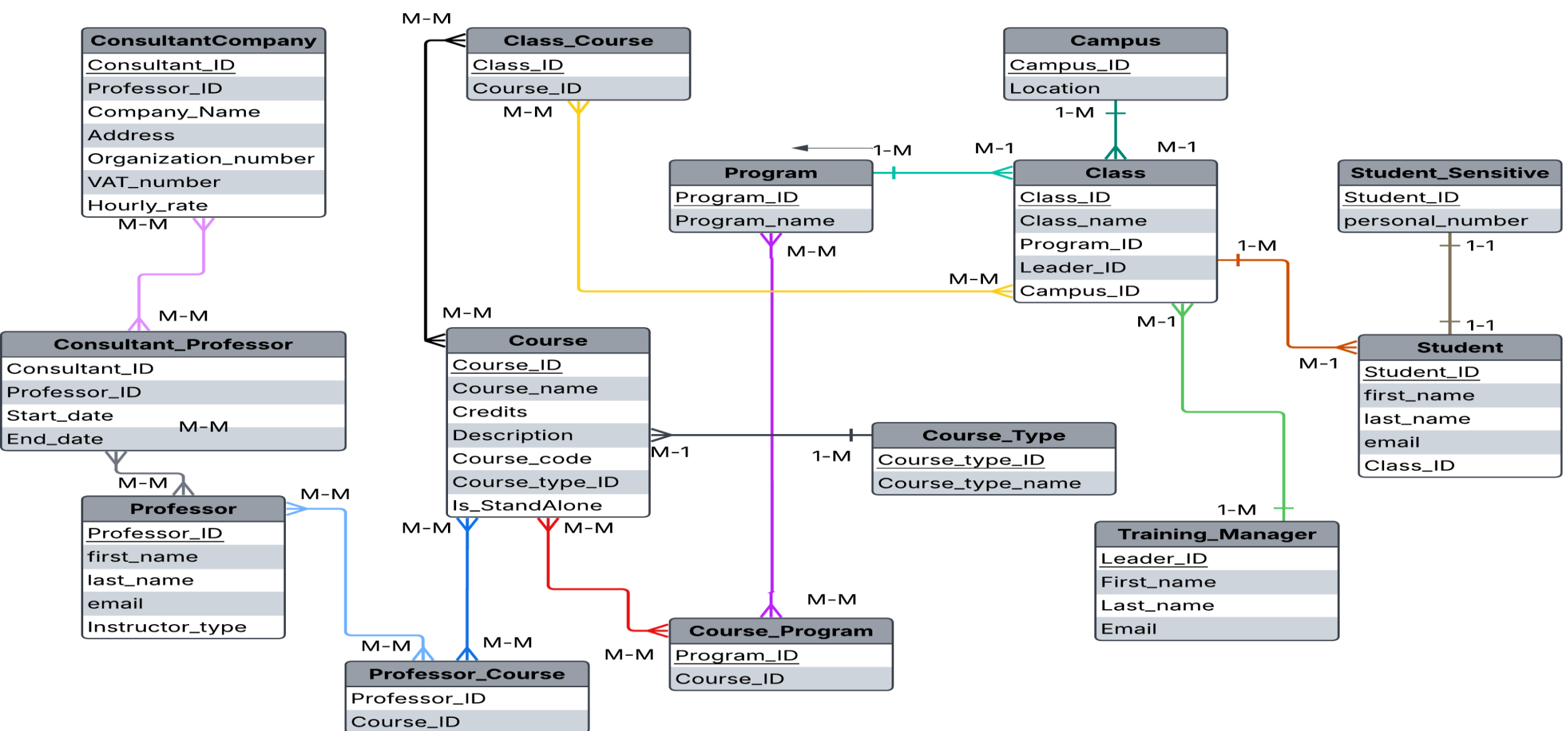
Excelfiler kan  
vara ineffektivt

# Från Excel till Databas: En Effektiv Lösning för YrkesCo



# Konceptuell Modell

- **Entiteter:** Studenter, Kurser, Program
- **Relationer:** Student  $\rightarrow$  Kurser
- **Struktur:** Dataintegritet & Effektiv design



Konzeptuell  
datamodell (ERD)

# Relationer i databasen



# Many-to-Many relationer

- **Many-to-Many-relationer:** En relation där flera rader i en tabell kan relatera till flera rader i en annan tabell.
- Exempel: En kurs kan tillhöra flera program, och ett program kan innehålla flera kurser.

# One-to-Many relationer

Exempel på One-to-Many relationer:

En student tillhör en klass men en klass har flera studenter



# One-to-One relationer

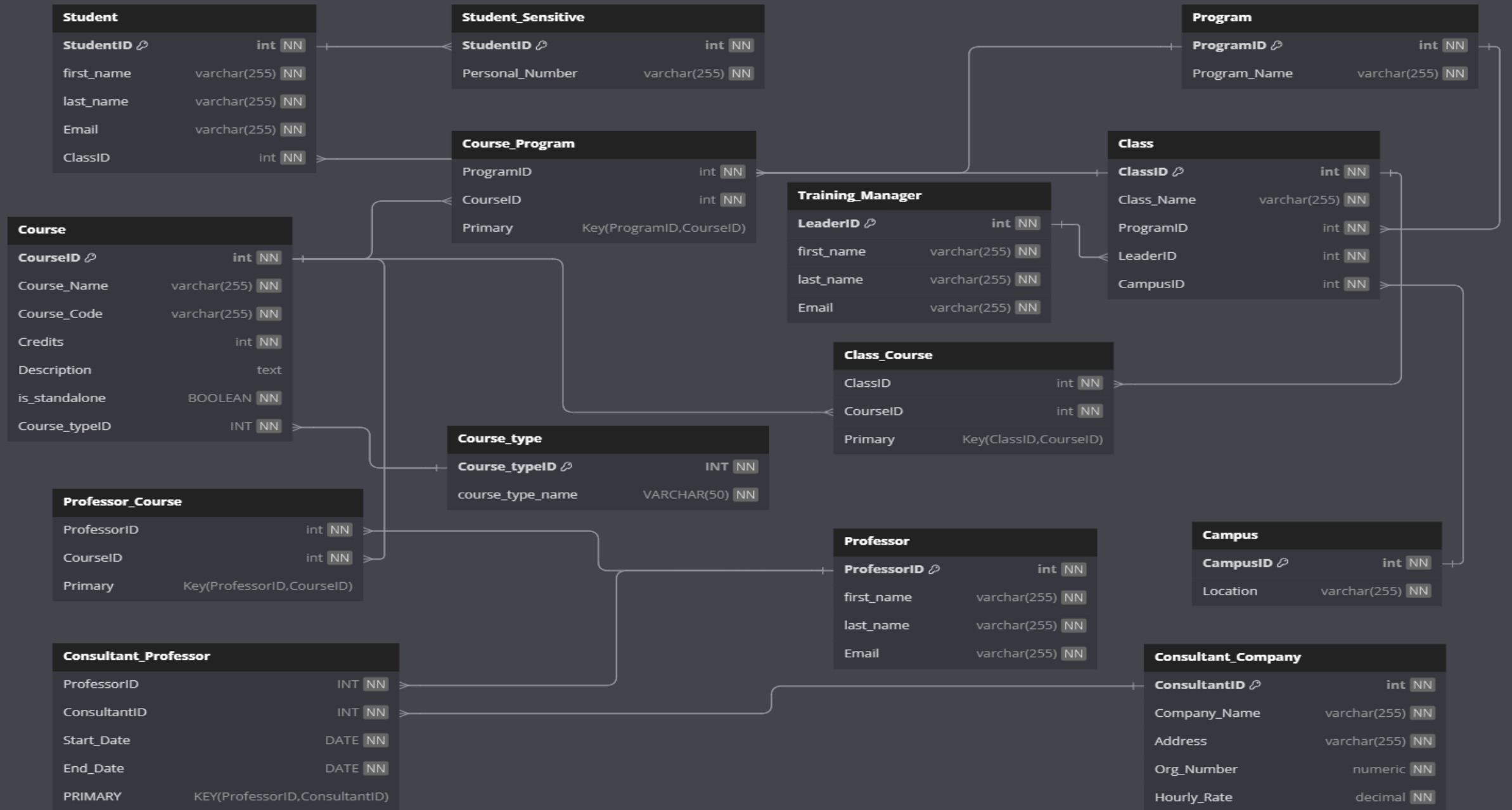
Student-Student\_Sensitive: Ett  
exempel på en-till-en-relation

# Logisk modell och normalisering

**Konceptuell modell** – Övergripande struktur och entiteter

**Logisk modell** – Definierade tabeller, datatyper och relationer

**Nycklar** – Primärnycklar och främmande nycklar för dataintegritet



# Normalisering

## Vad är Normalisering?

# 1NF, 2NF 3NF

- 1NF handlar om att eliminera upprepande grupper och säkerställa atomära värden.
- 2NF handlar om att säkerställa att icke-nyckelattribut är beroende av hela primärnyckeln.
- 3NF handlar om att säkerställa att icke-nyckelattribut endast är beroende av primärnyckeln.

# Vår Databas i 3NF

1NF: Atomära värden

2NF: Inget ofullständigt beroende

3NF: Inga transitiva beroenden

Minimera redundans och förbättra integritet



```

35 FROM
36     Class
37 JOIN
38     Training_Manager ON Class.Leader_ID = Training_Manager.Leader_ID
39 WHERE
40     Class.Class_Name = 'Class B';
41
42
43 --hämtar kursID, kursnamn, professors för och efternamn för alla kurser som ä
44
45 SELECT course.course_id, course.course_name, professor.first_name, professor.las
46 from course
47 left join professor_course on course.course_id = professor_course.course_id
48 left join professor on professor_course.professor_id = professor.professor_id
49 where course.is_standalone = TRUE;
50
51

```

	first_name character varying	last_name character varying	class_name character varying
1	Alice	Johnson	Class A
2	Charlie	Brown	Class A

1 row returned

	course_name character varying	first_name character varying	last_name character varying
1	Financial Accounting	Eva	Davis

1 row returned

	class_name character varying	first_name character varying	last_name character varying
1	Class B	Henry	Moore

1 row returned

	course_id integer	course_name character varying	first_name character varying	last_name character varying	is_standalone boolean
1	1010	Introduction to Data Science	David	Williams	true

List of relations			
Schema	Name	Type	Owner
public	campus	table	postgres
public	class	table	postgres
public	class_course	table	postgres
public	consultant_company	table	postgres
public	consultant_professor	table	postgres
public	course	table	postgres
public	course_program	table	postgres
public	course_type	table	postgres
public	professor	table	postgres
public	professor_course	table	postgres
public	program	table	postgres

# Implementing i PostgreSQL



PostgreSQL



# Möjlig vidareutveckling

Fler tabeller och detaljer

Prestandaoptimering

Användarroller & behörighet

Förbättrad datasäkerhet



Tack för er tid!

En modern databasstruktur som skapar ordning, minskar redundans och förbättrar datakvalitet – för en effektivare hantering av YrkesCo:s verksamhet.