



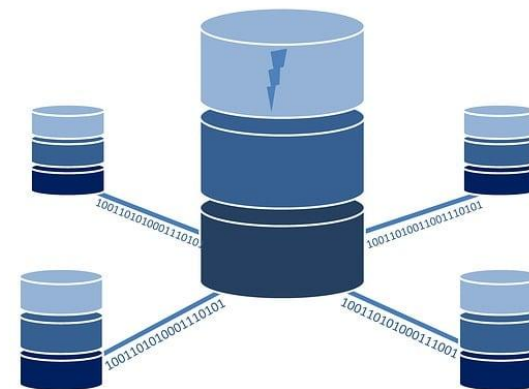
Databas för YrkesCo

Hej och välkomna! Idag kommer jag att prata om hur jag har byggt en databaslösning för YrkesCo, som hjälper till att hantera studenter, kurser och program på ett effektivt sätt.



Excelfiler kan
vara ineffektivt

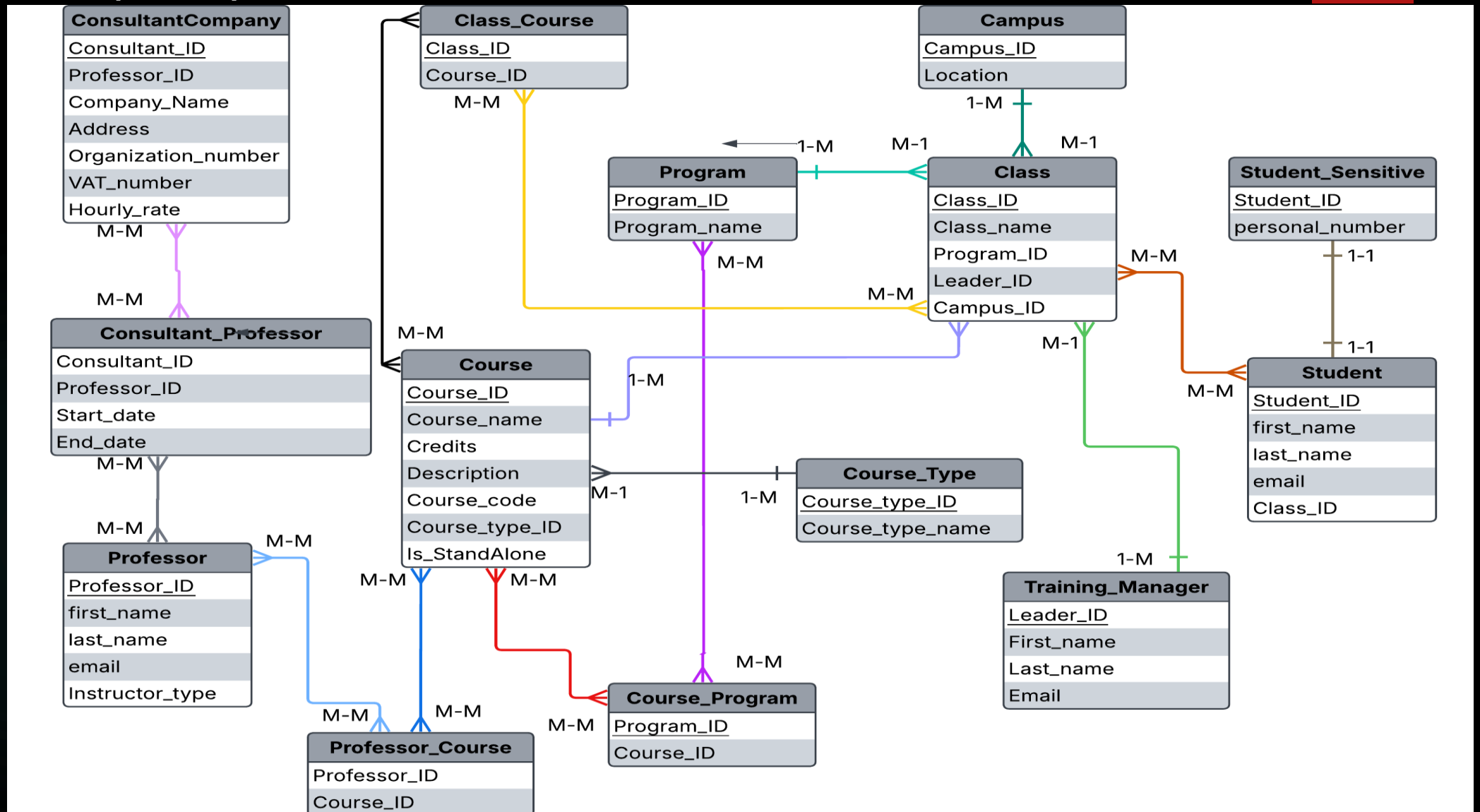
Från Excel till Databas: En Effektiv Lösning för YrkesCo



Konceptuell Modell

- **Entiteter:** Visar de viktigaste objekten som studenter, kurser och program.
- **Relationer:** Illustrerar hur entiteterna är kopplade, t.ex. en student kan registreras i flera kurser.
- **Struktur:** Säkerställer dataintegritet och en effektiv databasdesign.

Konzeptuell datamodell (ERD)



Relationer i databasen

Relationerna är viktiga!

De kopplar tabeller och säkerställer
dataintegritet.

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?

Normalisering är processen att organisera data i en databas för att minimera redundans och förbättra dataintegriteten.

1NF, 2NF 3NF

- ▶ 1NF innebär att varje kolumn i en tabell endast innehåller atomära värden, vilket betyder att varje cell innehåller ett enda värde och det inte finns några upprepande grupper.
- ▶ 2NF kräver att alla icke-nyckelattribut i en tabell är fullständigt funktionellt beroende av hela primärnyckeln.
- ▶ 3NF innebär att en tabell är i 2NF och att inga icke-nyckelattribut är transitivt beroende av andra icke-nyckelattribut.

Varför är vår databas i 3NF?

- Vår databas är i 3NF. Varje tabell uppfyller 1NF med atomära värden. 2NF säkerställs genom att alla icke-nyckelattribut är fullständigt beroende av primärnycklarna. Slutligen, 3NF uppfylls genom att inga icke-nyckelattribut har transitiva beroenden, vilket minimerar redundans och förbättrar integriteten. Brygggtabeller och separering av känslig data stödjer också denna normalisering.

```

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

PROBLEMS 28 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

dock

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



Implementering i PostgreSQL

Framtida utveckling

Vi kan vidareutveckla systemet genom att:

- ▶ Lägg till fler tabeller och detaljerad info om kurser och studenter.
- ▶ Optimera för prestanda.
- ▶ Implementera användarroller och behörigheter.
- ▶ Förbättra säkerheten för känslig data.



Effektivisering för en bättre
framtid.

Tack för er tid!