

Yrkesco

EN DATABAS FÖR ATT
ENKLARE KUNNA FÅ
ÅTKOMST TILL DATA
GÄLLANDE BLAND
ANNAT PROGRAM,
KURSER, LÄRARE OCH
STUDENTER.

- MILOU SANDÉN LINDROTH



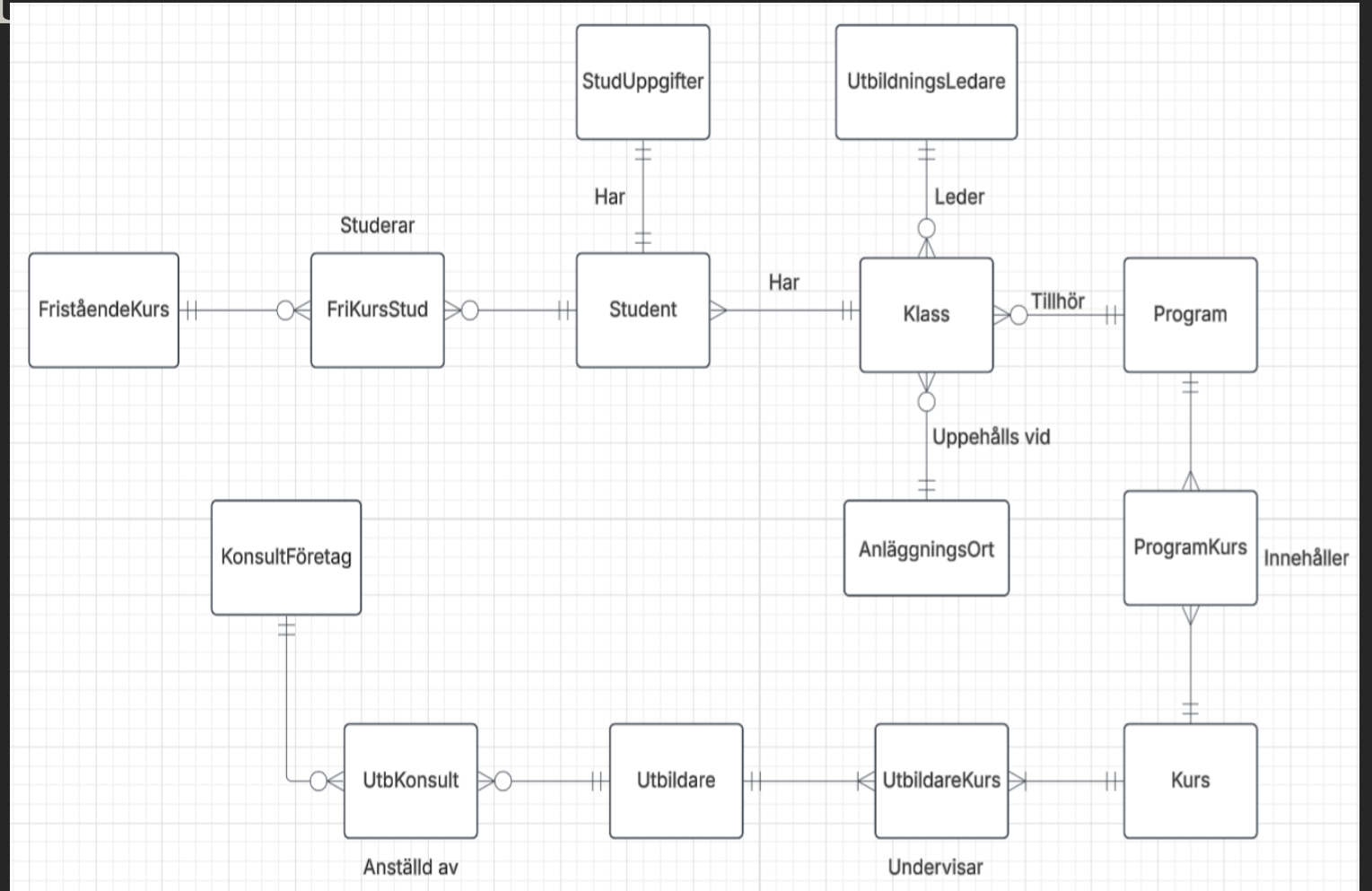
Conceptual model

* Min konseptuella modell som visar dom olika entiteterna, relationerna mellan entiteterna samt kardinaliteterna kopplade till dom.

* Jag har identifierat 10 grundentiteter med tillhörande 4 stycken bryggtabeller (composite entities).

* Relationerna mellan dessa entiteter är baserat på verkliga scenatier, tex: Ett program kan ha flera klasser, en klass har en utbildningsledare, varje kurs kan undervisas av flera utbildare etc.

* Exempel på kardinaliteterna: ett program inga eller flera klasser, medan en klass kan bara tillhöra ett program. Varje student har exakt ett personnummer. En utbildare kan undervisa en eller flera kurser, och utbildaren kan antingen vara anställd eller konsult från ett konsultföretag.

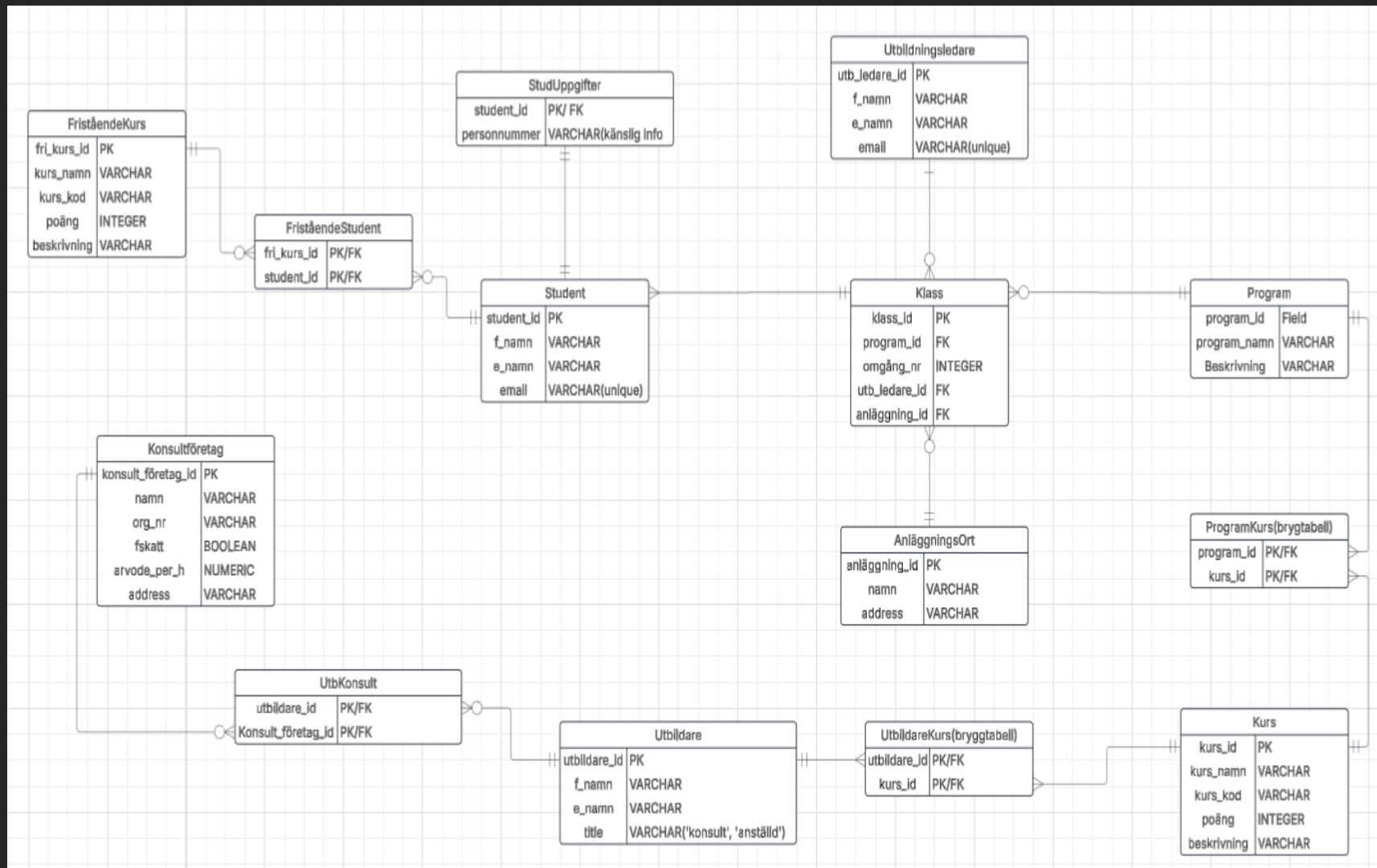


Logical model

* Detta är min logiska modell som även visar attributen i dom olika tabellerna, definierar datatyper, samt visar Primary keys och Foreign keys.

* Jag har till exempel valt primärnyckeln kurs_id för att lätt kunna koppla det till bryggtabellerna programkurs och utbildare kurs, för att få fram vilka kurser som finns i olika program, samt vem utbildaren är.

-> detta gör det enkelt att joina tabellerna och håller varje post unik, därmed lätt att spåra och även uppdatera med ny data om det skulle krävas.

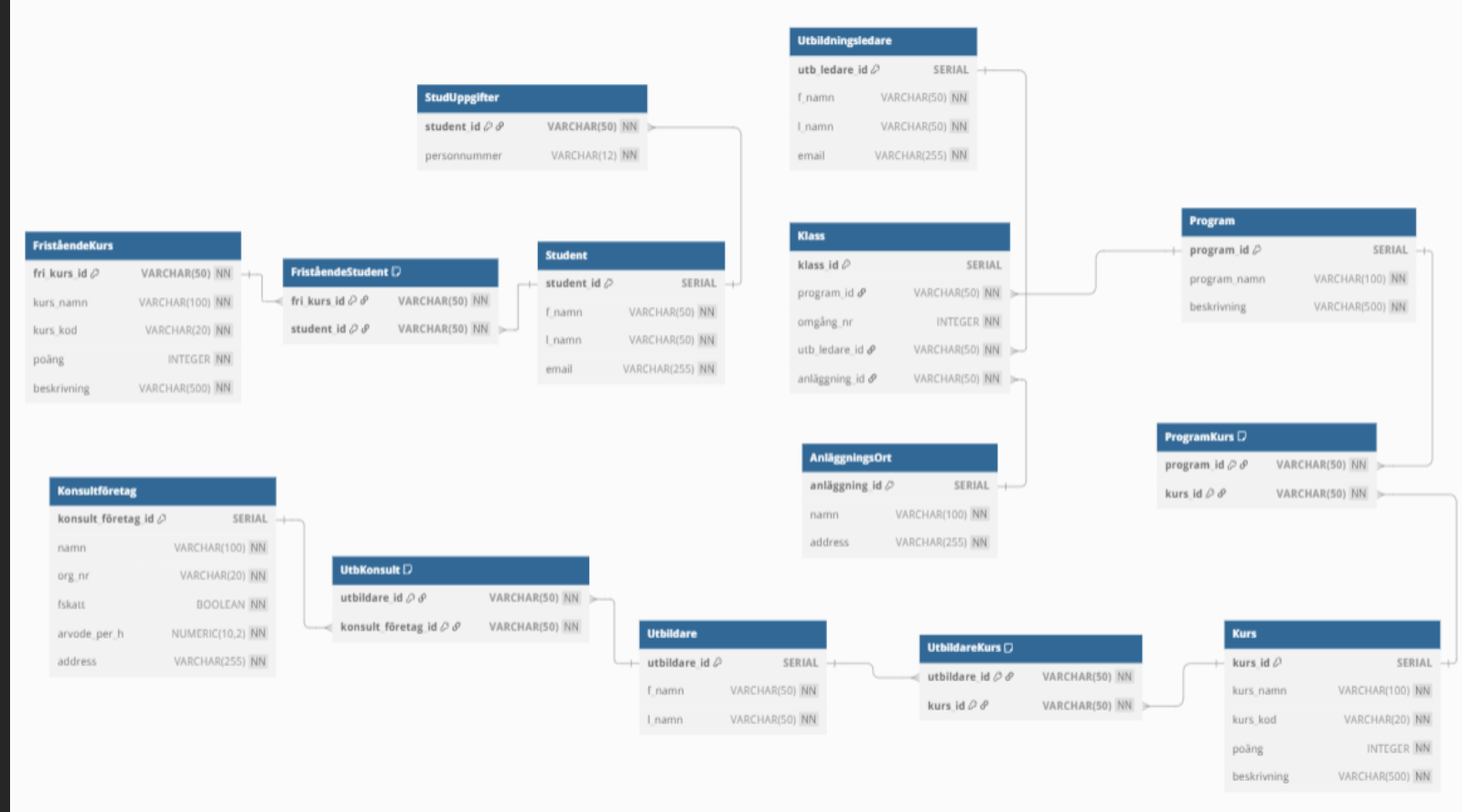


Physical model

Min fysiska modell som visar lite mer specific data och som nu är redo att implementeras till databasen.

* Varje tabell har en unik primary key, dessa används i andra tabeller som foreign keys för att koppla ihop datan.

* Detta är nödvändigt för att kunna analysera datan och göra joins, exempelvis: hur många studenter går i en viss klass= vilka kurser undervisas av utbildare från konsultföretag? etc.



Varför är min databas i 3NF?

Jag har byggt databasen i tredje normalformen. Det innebär att varje tabell bara innehåller data som hör ihop logiskt, och att alla beroenden är mellan nycklar. Det förhindrar insertion, update och deletion anomaly.

- * All data är 1 NF → det vill säga inga upprepade grupper, varje kolumn har bara ett värde.
 - * All data uppfyller 2NF → alla icke-nyckel attribut är fullt funktionellt beroende av hela primärnyckeln.
 - * all data uppfyller 3NF → inga transitiva beroenden. Med andra ord "non-prime attributes depend on the key, the whole key, and nothing but the key,
 - * Tillexempel min utbildare tabell, med primärnyckeln utbildare_id som har attributen namn och titel, dom är enbart beroende på primärnyckeln.
 - * Min bryggtabell ProgramKurs har foreign keys program_id och kurs_id, det är bara relationer i denna tabell och inga onödiga uppgifter som exempelvis kursnamnet igen.
-

Implementation - Postgres

* För att implementera min min modell så skapade jag tabllerna med en labb_setup sql- fil där jag använde mig av CREATE TABLE. För att lägga in fakedatan jag gjort för att säkerställa att databasen är uppbyggt på ett efttektivt och skalbart sätt så gjorde jag ett insert_data skript där jag la in det med hjälp av INSERT till dom redan skapade tabellerna. Jag skapade ett queries skript för att testa så att datan kom in korrekt, och att det var hållbara relationer mellan tabellerna.

Sammanfattningsvis

Med den här strukturen kan YrkesCo snabbt besvara frågor som rör kvalitet, planering och resursanvändning. Mitt system är flexibelt och kan enkelt utökas vid behov – om man till exempel vill lägga till distanskurser, statistik, nya anläggningsorter eller nya yrkesroller. Min databas är normaliserad, robust och lätt att underhålla.
