

Databaser - ER diagrammet

Programmering-B

Thomas Birk Abildgaard

February 2, 2026

HTX Frederikshavn

ER Diagrammet

Enhed(Entity)

Entity attributes

Relation (Relationship)

Cardinalitet (Cardinality)

Associative Entity

ER diagram → DB schema (SQL)

Opsummering

DATABASES



ER Diagrammet

Hvad er et ER-diagram?

ER står for **Entity-Relationship**

Et ER-diagram bruges til at **designer databaser**

Det viser:

Hvilke data vi gemmer

Hvordan data hænger sammen

ER-diagrammer bruges **før**, man laver databasen

Hvorfor bruge ER-diagrammer?

Giver overblik over systemet

Hjælper med at undgå fejl i databasedesign

Gør det lettere at:

- forklare databasen til andre (teams)

- omsætte design til tabeller (relationsmodel) SQL

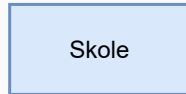
Enhed(Entity)

Enhed (Entity)

En enhed repræsenterer et objekt eller begreb, som har information, der skal gemmes i databasen.

Enheder er typisk navngivet i ental, f.eks. Kunde, Produkt, Ordre.

I ER-diagrammet repræsenteres enheden som et rektangel.



Entity attributes

Attributter i ER-diagrammer

Attributter beskriver egenskaber ved en enhed (entity), f.eks. Kunde_ID, Navn, Email.

Chen Notation:

Attributter vises som **ellipser** forbundet til enheden.

Primærnøgler er typisk **understreget**.

Sammensatte attributter kan vises opdelt i under-attributter.

Crow's Foot Notation:

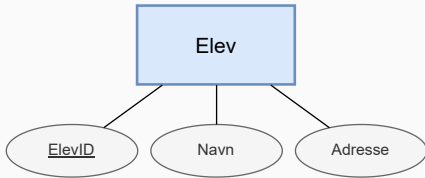
Attributter skrives som **tekst inde i enheden**.

Primærnøgler markeres ofte med PK, fremmednøgler med FK.

Modellen ligger tæt på det endelige SQL-skema.

Attributter

CHEN



CROW

Elev	
PK	<u>ElevID</u>
	Navn
	Adresse

Relation (Relationship)

Relation (Relationship)

En relation beskriver, hvordan to eller flere enheder er forbundet.

Eksempel:

Elev følger Fag

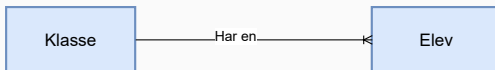
Lærer underviser i Fag

Relationer kan være **en-til-en**, **en-til-mange** eller **mange-til-mange**.

I ER-diagrammet repræsenteres relationen således.



CROW



En-til-en (1:1)

En-til-mange (1:M)

Mange-til-mange (M:N)

For hver relation skal du kunne sige:

“En A er forbundet med $[min..max]$ B ”

“En B er forbundet med $[min..max]$ A ”

Cardinalitet (Cardinality)

Cardinalitet (Cardinality)

Cardinalitet beskriver antallet af instanser, der kan være forbundet med en relation.

De vigtigste typer af kardinalitet er:

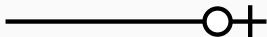
1:1 (One to One): Hver instans af en enhed er forbundet med én instans af den anden enhed.

1:N (One to Many): En instans af den første enhed kan være forbundet med flere instanser af den anden enhed.

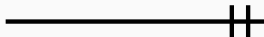
M:N (Many to Many): Flere instanser af begge enheder kan være forbundet med hinanden.

I Crows Foot Notation vises dette ved hjælp af "fødder" (hvor en fod repræsenterer "mange").

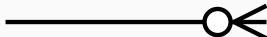
Sådan læses Crow's Foot notation



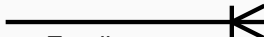
Nul eller en
(optional)



En og kun en
(mandatory)



Nul eller mange
(optional)



En eller mange
(mandatory)

Uspecificeret minimum i Crow's Foot



One og Many har ingen angivet nedre grænse.

Konsekvens

Det er uklart, om relationen er **obligatorisk** eller **valgfri**

Hvorfor kan det være nyttigt?

Forretningsregler er endnu ikke fuldt kendte (designfase)

Minimum er ikke vigtigt på modelleniveau

Beslutningen kan udskydes til konfiguration eller database

Uspecificeret minimum i Crow's Foot



One og Many har ingen angivet nedre grænse.

Konsekvens

Det er uklart, om relationen er **obligatorisk** eller **valgfri**

Hvorfor kan det være nyttigt?

Forretningsregler er endnu ikke fuldt kendte (designfase)

Minimum er ikke vigtigt på modelleniveau

Beslutningen kan udskydes til konfiguration eller database

- *Undgå disse notationer hvor det er muligt!.*

Associative Entity

Associative Entity

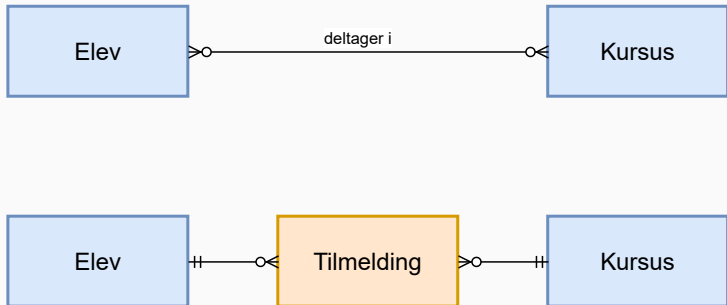
En **Associative Entity** bruges til at håndtere mange-til-mange (M:N) relationer.

Når en M:N-relation findes mellem to enheder, *oprettes en ekstra enhed* (associative entity), som indeholder de nødvendige referencer til begge enheder.

Eksempel: Hvis en Studerende kan tage flere Kurser, og et Kursus kan have flere Studerende, vil en Studerende_Kursus enhed blive oprettet.

Denne enhed (associative entity) vil have en **sammensat primærnøgle**, som består af de primære nøgler fra de to relaterede enheder.

Associative Entity



ER diagram $\rightarrow$ DB schema (SQL)

Entiteter bliver til tabeller

Attributter bliver til kolonner (rel.datatype)

Primærnøgler og **fremmednøgler** forbinder tabeller

ER-diagrammet er **grundlaget** for SQL-databasen

Opsummering

ER-diagrammer bruges til at designe databaser

Vigtige begreber:

- Entiteter

- Attributter

- Relationer

- Kardinalitet

- Associativ Entity

Godt ER-design giver bedre databaser

Det er ofte en god ide at:

Normalisere ER-diagrammet under designfasen, så du skaber en logisk struktur, der allerede er i 1NF, 2NF, og 3NF.

Opret SQL-tabeller baseret på det normaliserede ER-diagram.

Lav et ER-diagram for:

Elever

Klasser

Fag

Overvej:

Hvilke relationer findes der?

Hvad er primærnøgler?