

defineres mange forskellige grader, idet værdien på begge sider af relationerne kan være både 0, 1, mange eller en helt konkret værdi.

## 6.8.2 ER-diagrammer

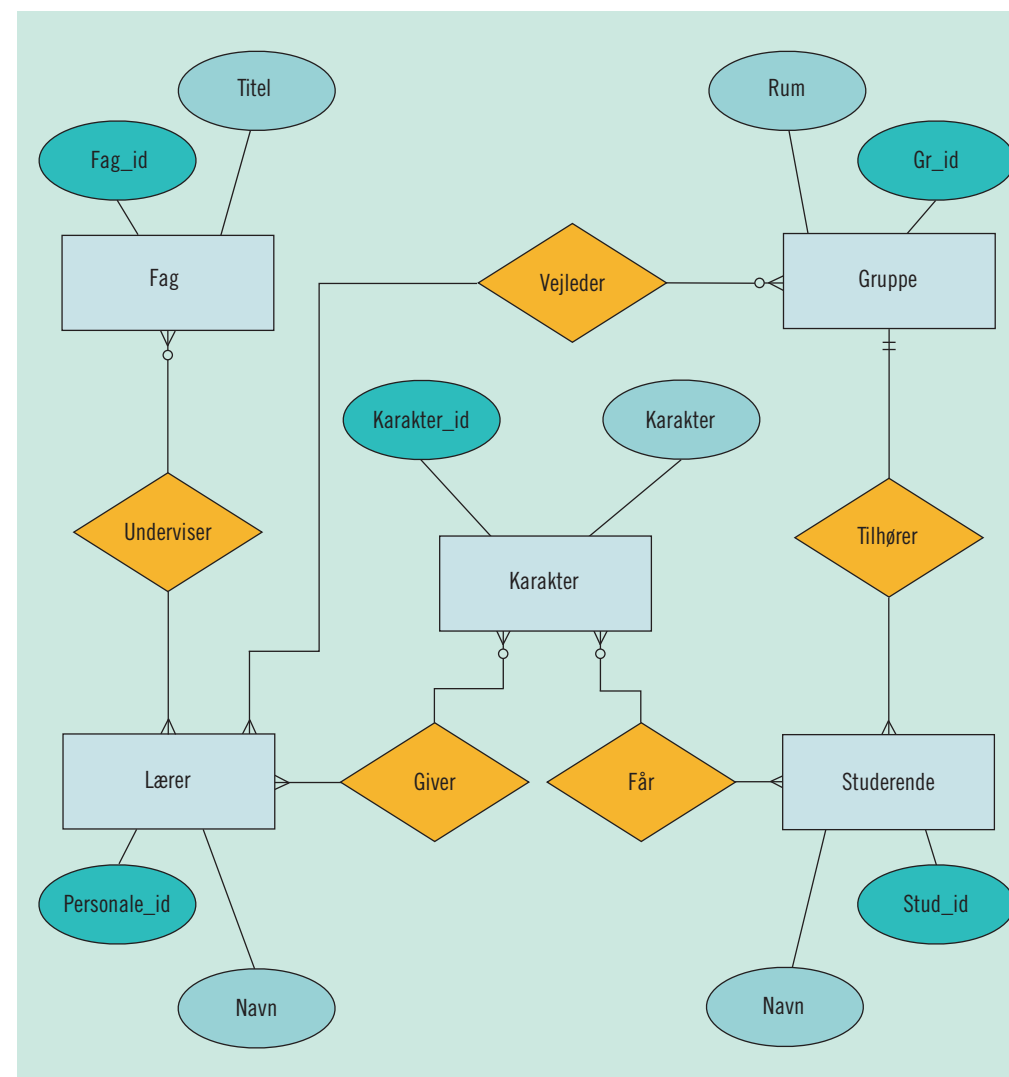
Når en database skal designes og kommunikeres til andre, er første skridt at modellere den konceptuelle datamodel. I den relationelle databasemodel kan dette gøres vha. et **Entitets-Relations Diagram** (forkortet et **ER-diagram**). Disse diagrammer er vigtige for at kunne forstå sammenhængen i databasen. Her opbygges databasemodellen vha. entiteterne og relationerne mellem disse. En entitet er som beskrevet tidligere en unik forekomst, og derfor er det **entitetsklasserne** (en generalisering af entiteter), der bruges i forbindelse med datamodelleringen. Ellers ville en databasemodel ikke kunne anvendes generelt, men kun være gældende for bestemte situationer. Entitetsklassen kan desuden beskrives vha. de tilhørende attributter.

Der er tre grundlæggende symboler, der bruges i ER-diagrammerne (se figur 6.5). Det første symbol er rektanget, der beskriver entitets-klassen. I tilknytning til entitets-klasserne tegnes attributterne som ovaler, og disse forbindes til entitetsklasserne med en

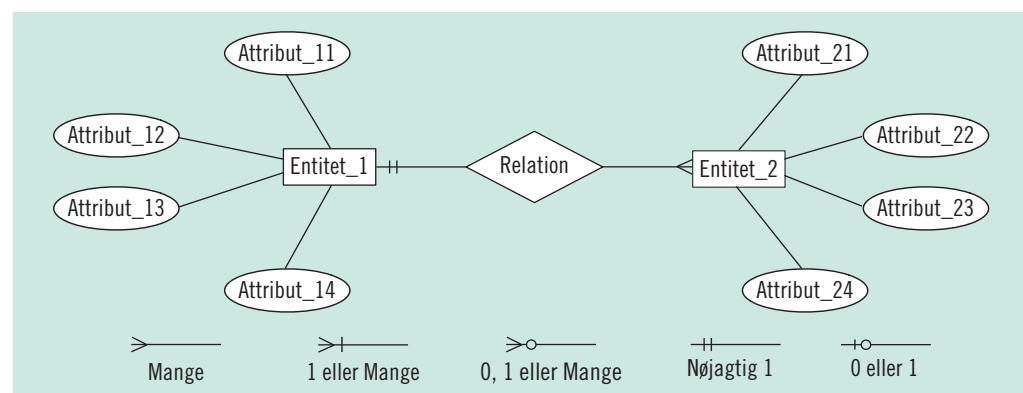
enkelt streg. Attributterne kan have en funktion som primær-nøgle, hvilket vises som en understregning af navnet. I de tilfælde, hvor attributten er beregnet på baggrund af fx andre attributter, vil den blive vist med en enkelt stiplede linie. Det kan fx være et areal, der skal beregnes ud fra andre attributter som fx koordinater, der igen er beregnet på baggrund af observationer. Hvis der tilføjes/rettes i observationerne, beregnes nye koordinater og dermed også nye arealer. Det sidste symbol, der bruges i ER-datamodelleringen, er romben, som betegner relationerne. Foruden de tre typer symboler, påføres ER-diagrammet også oplysninger om graden af relationerne. Dvs. at man direkte fra diagrammet kan læse om relationen er fx en én-til-mange eller en én-til-én relation. Det vises med forskellige symboler, der hvor relationen rammer entitetsklassen. Figur 6.6 viser et eksempel på et forholdsvis simpelt ER-diagram.

## 6.8.3 Tabeller, poster, felter og nøgler

På det logiske niveau arbejdes der med **tabeller, poster, felter** og **nøgler** i datamodelleringen. Her udgør hver enkelt post i en tabel det, som i den konceptuelle datamodel



► Figur 6.6 Eksempel på ER-diagram. Farverne er ikke standard, men gengives her for overskuelighedens skyld.



► Figur 6.5 Sådan modelleres de forskellige elementer fra ER-diagrammet.

repræsenteres af entiteter. På samme måde er attributterne i den konceptuelle database-model blevet til felter i den logiske database-model. For hver attribut dannes der i tabellen en ny kolonne og dermed et nyt felt (se figur 6.7). Hvert felt kan for hver post indeholde én værdi. Felterne har en meget vigtig funktion i databasen, fordi de er databærende, og derfor er det nødvendigt at modellere dem korrekt. Dette indebærer, at datatypen

for feltet skal fastlægges, og at der dedikeres primærnøgler og fremmednøgler til de relevante felter.

Et felt kan antage mange forskellige datatyper: en tekst, et tal, en dato eller binære værdier. De mest anvendte datatyper vil blive gennemgået i det efterfølgende afsnit, men inden da vil principperne for nøglefelterne i databasen blive gennemgået.