# Objektorientering VS Relationelle databaser

Data i relationelle databaser ligner meget den opbygning der findes i Objektorienteret og det betyder at hvis vi sammenligner dem step for step, vil vi se at det nærmest blot er navne der ændre sig:

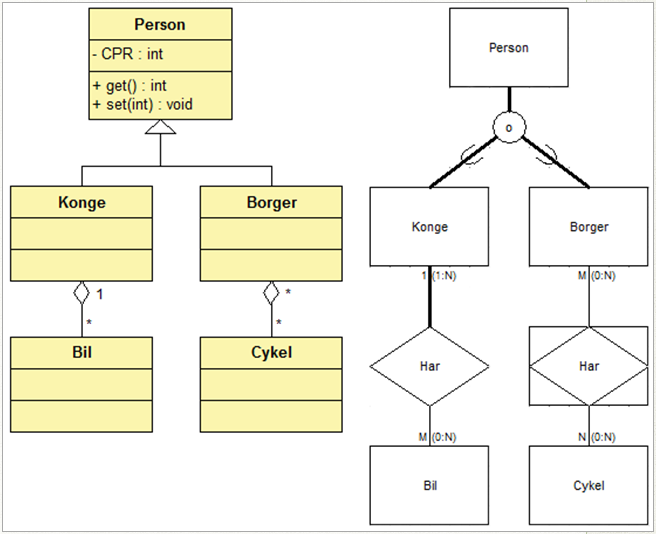
|  |  |
| --- | --- |
| **ObjektOrientering** | **Relationel Database** |
| Klasser | Entiteter |
| Data i attribut i objekter | Data i attribut i tabeller |
| Ændre data via metoder: get(), set() | Ændre data via DML(data Manipulation Language) CRUD (Create, Read, Update, Delete) |
| Relationer dannes mellem klasser med pointere og referencer | Relationer mellem entiteter dannes på baggrund af foreignkeys |
| Beskrives med et Klassediagram | Beskrives med et ER Diagram |

I en relationel model er data gemt i tabeller, hvor hver tabel har kolonner og rækker. Fremmednøgler anvendes til at til at lave relationer mellem andre tabeller. Heri findes der forskellige typer relationer: Many-to-many, one-many, one-to one og n-ary. Der regler hvorledes man overholder integriteten af data – bl.a. vha. normaliseringen.

I en objekt-orienteret model anvender man objekter til at gemme og manipulere data på. De tre faser til modellering er analyse, design og implementering. Relationer mellem objekter kan ske via komposition (”har en”-relation), association (”anvender”-relation) og arv (”er en”-relation).

Klasser i den objekt-orienteret model kan mappes til en tabel, hvor hver attribut i klassen er kolonnen i tabellen.

Det er muligt at mappe et klasse objekt til relationel database. Heri bestemmes der hvilke attributter der skal være i OO-model og hvilke relationer et objekt har til andre objekter. Disse attributter og relationer kan oftest direkte konverteres til entiter med relationer imellem bestemt af fremmednøgler. Figur 1 giver et godt overblik over dette.



Figur : forskelle på diagrammerne

**Entity Framework**

Lader os lave en objektorienteret model, som automatisk bliver lavet til en relationel database uden SQL-statements