## Valg til Entiteterne i databasen:

Planen med den første udgave af databasen var, at den skulle indeholde en enkel tabel, hvor man kunne se varens navn i rækkerne og de forskellige forretninger hen ad kolonerne, og på den måde finde prisen for varen i den enkelte forretning. Dette viste sig at have nogle problemer, da vi skulle lave klassen Product med en variabel for hver forretning, som vi kunne gemme prisen i. Det resulterede i, at vores database ikke ville være åben for udvidelser, hvilket er ineffektivt, når forretninger kan åbne og lukke, og forretningerne får nye varer i deres sortiment. Der er en mange til mange relation imellem forretningerne og produkterne, da en forretning kan have mange forskellige produkter, og ét produkt kan blive solgt i mange forskellige forretninger. Der er derfor valgt at lave en klasse til relationen, da der er behov for en variabel til at indeholde prisen for produktet i den specifikke forretning. Derfor blev der oprettet 3 entiteter i databasen; Store, Product og HasA.

Klassen HasA indeholder en pris samt relationer til produktet og forretningen. Disse tre klasser gør, at databasen kan håndtere, at en forretning begynder at sælge et nyt produkt, samt at en ny forretning åbner. Måden, det er implementeret på, er ved, at Product indeholder en liste af HasA. Dette betyder, at for produkterne og forretningerne er der kun én instans, for hvert produkt eller forretning der er, og så er der en HasA imellem hver af de forretninger, som sælger ét produkt, og de kobles dermed sammen på den måde. På samme måde indeholder Store også en liste af HasA, der angiver, hvilke produkter der sælges i den forretning.

For at forretningerne ikke kan ændre prisen, eller hvilke produkter der sælges ved andre forretninger, er der lavet et login til hver forretning, og alle logins gemmes i databasen. Denne entitet har et brugernavn, kodeord, og en reference til den forretning, de har kontrol over. Så når der logges ind for en forretningsbestyrer, så har personen kun mulighed for at ændre, hvordan hans forretning skal fremstå. Derudover er der lavet så Administratoren, som kan tilføje og slette forretninger, også har et login. Dette login er specielt i forhold til de andre, da det giver adgang til administrationsdelen af forretningerne. Derfor blev Administratorens login lavet med en forretning, der hedder Admin, og på den måde tages der i programmet højde for, hvilken retning administrationsprogrammet skal tage efter login.