

DB

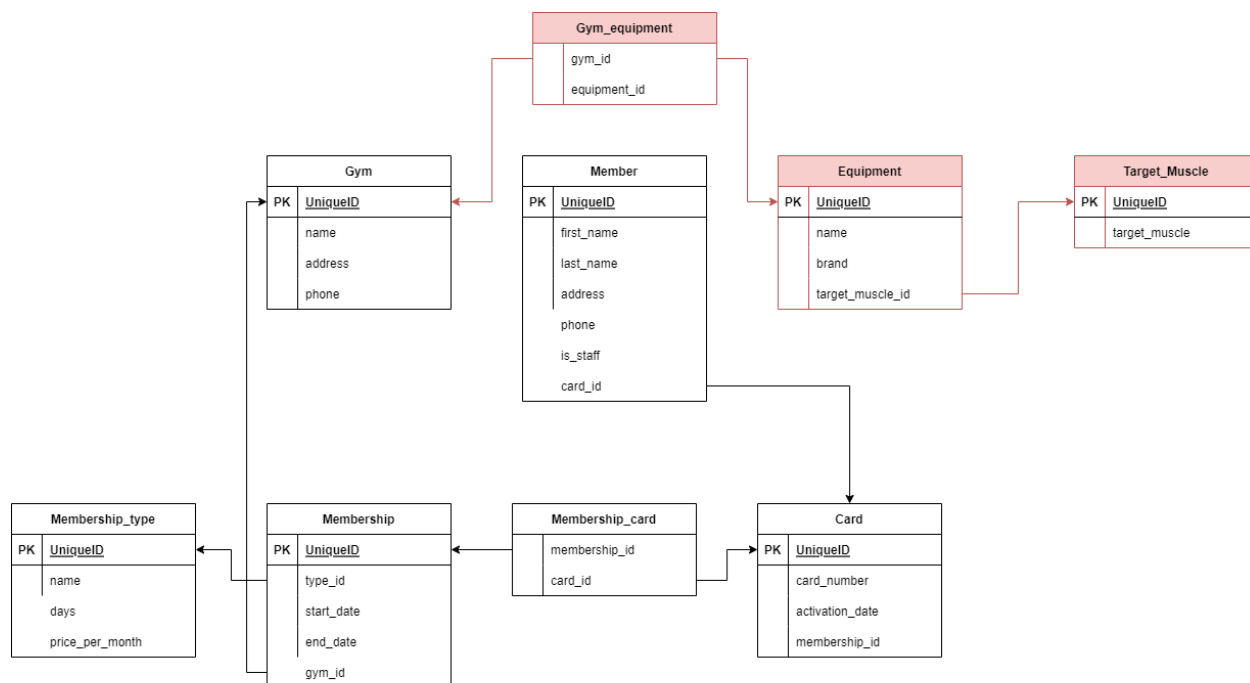
# Gym System

## Projektrapport

### Introduktion

Detta är en rapport av projektet Gym System. Ett system som ska kunna implementeras av gym för att sköta hanteringen av medlemmar och medlemskap. Kommer främst dyka in på databas delen av projektet men delar av programmeringen kommer även tas upp.

### Kopplingsdiagram (databas)



Rödmarkerade tabeller är ej implementerade i nuvarande version av programmet.

### Databasdesign

Relationsdatabasen är designad så att en *Member* kan inneha max ett *Card*. Alltså en så kallad One-to-One relation. Detta kortet (*Card*) kan i sin tur tilldelas flera *Membership*:s av

---

olika typer, men varje *Membership* kan tilldelas olika kort. (Many-to-Many). Varje *Membership* kan endast vara av en *Membership\_type* och endast tillhöra ett *Gym*. Tanken med denna designen är att det endast ska krävas ett kort per medlem oavsett hur många gym medlemmen väljer att träna på.

## Processen

Idén var tydlig från början. Det ska kunna finnas flera gym med flera medlemmar. Det viktiga var att alla gym inom en viss kedja/företag ska kunna använda sig av ett gemensamt datasystem.

I en tidig design av databasen krävdes det ett kort per medlemskap och gym. Detta var inte praktiskt och tanken kom till att endast ett kort ska behövas per medlem. Den tidigare medlemskaps tabellen fick därför delas upp i två stycken tabeller *Card* och *Membership*, med en *Many-To-Many* relation.

## Förbättringspotential

I nuvarande version av databasen används inte *Cascade Delete*, vilket komplicerar det på programmerings sidan av projektet. Detta innebär att i nuvarande systemet behövs flera SQL queries behöver göras från programmet för att ta bort en medlem och dess relationer i databasen. Med *Cascade Delete* kan allt detta göras automatiskt i databasen.

Tanken är också att implementera hanteringen av gymmets utrustning i gym systemet också. Detta har inte gjorts än.