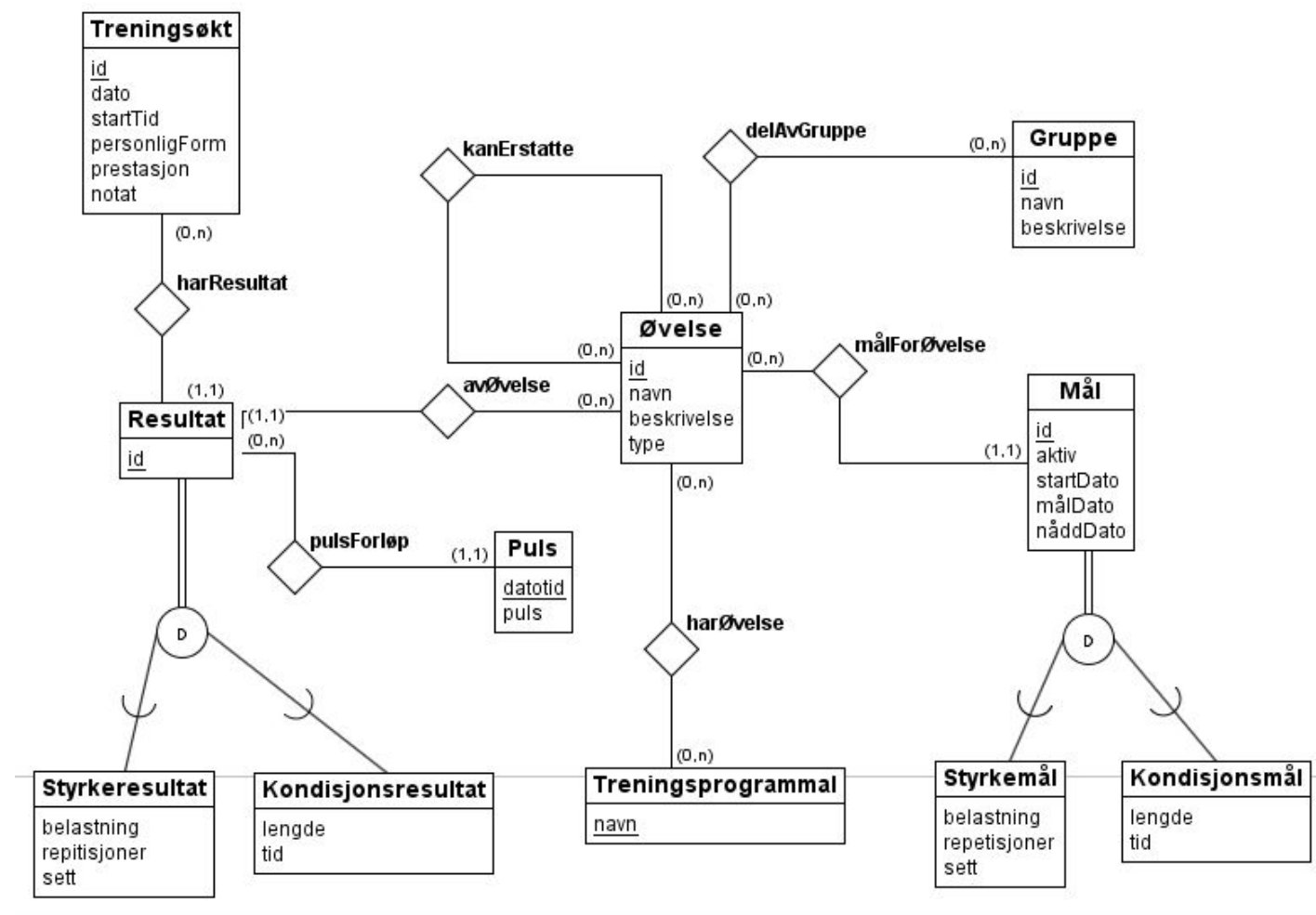


Prosjekt i TDT4145 - Innlevering 1

Gruppe 64:
Christer Hølestøl
Mats Sæter
Jonas Einar Thorsen
Kristian Trandem

ER-diagram:



Oppgave 1 a) b:

Vi har valgt å rette oss mot trening ved treningscenter.

Noen øvelser vi vil lagre er: knebøy(8 rep, 3 sett, 100 kg),
benkpress(8 reps, 3 sett, 100 kg),
beinpress(8 reps, 3 sett, 150 kg),
markløft(8 reps, 3 sett, 120 kg),
nedtrekkmaskin(8 reps, 3 sett, 110 kg) og
roing(20 km/40 min).

Temperatur og værforhold er for oss uinteressante.
En trening kan ha form: god; prestasjon: god.

Et evt. notat kan være: "Treningen i dag gikk knirkefritt, bortsett fra benkpressen som var litt tung. Der klarte jeg bare 5 reps på siste settet."

1.

Vi har en entitetsklasse som heter Treningsøkt hvor vi har med tilhørende data, som id, dato, og tid. Brukeren får her mulighet til å legge til egne notat der vedkommende kan skrive hvordan treningen har gått.

Vi har også med i resultatet hvilke(n) øvelse som ble utført og avhengig av hva slags øvelse det var er det med egne styrke- eller kondisjonsresultat.

2.

Vi har egne entitetsklasser "Treningsøkt" og "Øvelse" hvor brukeren kan legge inn hva hen ønsker. Under øvelse kan brukeren sette seg mål og se tidligere mål, samt informasjon om når målene skal være innfridde og om målene evt. er nådd.

3.

Vi lar brukeren registrere resultatet fra treningene sine, samt ha tilgang til sine allerede nådde mål.

4.

Man vil ikke kunne finne denne informasjonen direkte i databasen, men all nødvendig data er lagret. Datoen fins i entitetsklassen "Treningsøkt", selve resultatet ligger i "Resultat" og hvilken øvelse som ble gjort lagres i "Øvelse". Tidligere og nåværende mål er lagret i "Mål". Java-programmet vil bruke denne informasjonen til å gjøre utregningene som trengs.

5.

Her har vi laget en egen entitetsklasse kalt Treningsprogrammal.
Der kan brukeren lagre maler for både tidligere treninger, samt helt nye.

6.

Gps-data er ikke relevant da vi har tatt for oss trening på treningsstudio.
Vi har derimot en egen entitetsklasse kalt Puls, hvor puls blir registrert opp mot resultat.

7.

Dette kan programmet gjøre. Vi lar brukeren lagre notater for hver trening.

8.

Vi har en entitetsklasse for øvelser og en for grupper. Resten
blir opp til programkoden.

1 b) b:

For alle sterke klasser som ikke har spesialisering har vi laget en egen relasjon.

For alle mange-til-mange-relasjoner (HAR_OVELSE, DEL_AV_GRUPPE) har vi laget en egen relasjon med foreign keys til "deltagerne".

Der vi har en-til-mange-relasjoner har vi lagt inn en foreign key på den siden det kun finnes en av.

Vi har lagt inn en rekursiv relasjon (KAN_ERSTATTE) ut fra OVELSE.

Siden at det er en total spesialisering og disjunkt, har vi valgt å implementere enhver subklasse som en egen relasjon.

Nøkkel og attributter fra superklasser blir lagret i hver subklasse.

Vi har lagt inn onDelete- og onUpdate-informasjon, slik at der det er en-relasjon, må nødvendigvis denne tabellen slettes eller oppdateres når tabellen den er avhengig av endres/slettes.