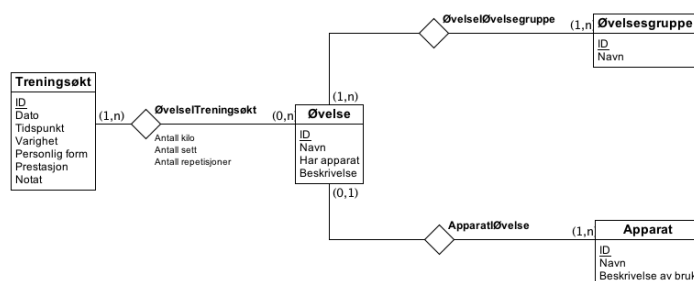


# Innlevering 1: Konseptuell databasemodell

Eskild Trøen, Amalie Hennie, Sindre Sivertsen, Simen Holmestad

March 6, 2019

## A) ER-diagram



ER-diagram

### Antagelser:

- Databasen er treningsdagboken til én bruker

- En øvelse er en generell øvelse og ikke spesifikk for en treningsøkt
- En øvelse er en apparatøvelse hvis HarApparat er true.
- Hvis en øvelse ikke er en apparatøvelse kan attributtene i ØvelseITreningsøkt settes til null

## B) Relasjonsdatabasemodell

- Treningsøkt( TreningsøktID, Dato, Tidspunkt, Varighet, PersonligForm, Prestasjon, Notat )
- Øvelse( ØvelseID, Navn, Beskrivelse, HarApparat, ApparatID )
- ØvelseITreningsøkt( TreningsøktID, ØvelseID Kilo, AntallSett, AntallReps )
- Apparat( ApparatID, Navn, HvordanBruke )
- Øvelsesgruppe( ØvelsesgruppeID, Navn )
- ØvelseIØvelsesgruppe( ØvelsesgruppeID, ØvelseID )

Vi valgte å modellere Kilo, AntallSett og AntallReps på relasjonen mellom Øvelse og Treningsøkt fordi øvelsene da kan være generelle øvelser og at man slipper å opprette nye øvelser for hver treningsøkt. Når vi kobler økt mot øvelse vil vi da også få med hvor bra det gikk på de forskjellige øvelsene akkurat denne treningsøkten.

## C) Hvordan oppfylle krav

Oppgave: For hvert nummerert punkt i kravspesifikasjonen skal det kort forklares hvordan modellen deres oppfyller kravet til en slik funksjonalitet.

### 1. Registrering

Vi har tabeller i SQL som kan registrere apparatoer, øvelser og treningsøkter med tilhørende data.

### 2. Få opp informasjon om n siste treningsøkter

Her kan vi selecte treningsøkter på ID så vi får ut de n siste og kan vise informasjon om dette.

### 3. Resultatlogg av øvelser i tidsintervall

Ved å joine Øvelse og Treningsøkt på gitt dato (og evt. tidspunkt) på dato (og tidspunkt) kan vi få ut øvelsene gjennomført i det gitte tidsrommet. Man kan se ulik prestasjon i antall kilo, sett og reps i relasjonen mellom Treningsøkt og Øvelse.

### 4. Finne øvelser i samme øvelsesgruppe

Vi har testdata lagt inn i databasen som kan brukes til å teste med. Siden alle øvelser må ha lagt inn hvilke(n) øvelsesgruppe den er i, kan man opprette en øvelsesgruppe og sjekke om øvelsen har den gitte øvelsesgruppa.

### 5. Valgfritt use case

Få informasjon om totalt antall kilo løftet, repetisjoner og sett utført av brukeren. Gjennom ØvelseITreningsøkt-tabellen har vi tilgang til alle øvelser utført av brukeren og attributtene Kilo, AntallSett og AntallReps. En summering over radene i de respektive kolonnene vil gi ønske resultat.

## D) SQL-script

```
DROP TABLES IF EXISTS Treningsøkt, Øvelse, Apparat, ØvelsesGruppe,
ØvelseITreningsøkt, ØvelseIØvelsesgruppe;
CREATE TABLE Treningsøkt (
    TreningsøktID INTEGER NOT NULL AUTO_INCREMENT,
    Dato DATETIME NOT NULL,
    Varighet TIME NOT NULL,
    PersonligForm INTEGER NOT NULL,
    Prestasjon INTEGER NOT NULL,
    Notat TEXT NOT NULL,
    CONSTRAINT Treningsøkt_PK PRIMARY KEY (TreningsøktID)
);
CREATE TABLE Apparat (
    ApparatID INTEGER NOT NULL AUTO_INCREMENT,
    Navn VARCHAR(30) NOT NULL,
    HvordanBruke TEXT NOT NULL,
    CONSTRAINT Apparat_PK PRIMARY KEY (ApparatID)
);
CREATE TABLE Øvelse (
    ØvelseID INTEGER NOT NULL AUTO_INCREMENT,
    Navn VARCHAR(30) NOT NULL,
    Beskrivelse TEXT NOT NULL,
    HarApparat BOOLEAN NOT NULL,
    ApparatID INTEGER NOT NULL,
    CONSTRAINT Øvelse_PK PRIMARY KEY (ØvelseID),
    CONSTRAINT Øvelse_FK FOREIGN KEY (ApparatID)
REFERENCES Apparat (ApparatID)
ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE ØvelsesGruppe (
    ØvelsesgruppeID INTEGER NOT NULL AUTO_INCREMENT,
    Navn VARCHAR(30) NOT NULL,
    CONSTRAINT ØvelsesGruppe_PK PRIMARY KEY (ØvelsesgruppeID)
);
CREATE TABLE ØvelseITreningsøkt (
    TreningsID INTEGER NOT NULL AUTO_INCREMENT,
    ØvelseID INTEGER NOT NULL,
    Kilo FLOAT(24),
    AntallSett INTEGER,
```

```

        AntallReps INTEGER,
        CONSTRAINT ØvelseITreningsøkt_PK PRIMARY KEY (TreningsID , ØvelseID),
        CONSTRAINT ØvelseITreningsøkt_FK1 FOREIGN KEY (TreningsID)
REFERENCES Treningsøkt (TreningsøktID)
ON UPDATE CASCADE ON DELETE CASCADE,
        CONSTRAINT ØvelseITreningsøkt_FK2 FOREIGN KEY (ØvelseID)
REFERENCES Øvelse (ØvelseID)
ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE ØvelseIØvelsesgruppe (
        ØvelseID INTEGER NOT NULL AUTO_INCREMENT,
        ØvelsesgruppeID INTEGER NOT NULL,
        CONSTRAINT ØvelseIØvelsesgruppe_PK PRIMARY KEY (ØvelseID , ØvelsesgruppeID),
        CONSTRAINT ØvelseIØvelsesgruppe_FK1 FOREIGN KEY (ØvelseID)
REFERENCES Øvelse (ØvelseID)
ON UPDATE CASCADE ON DELETE CASCADE,
        CONSTRAINT ØvelseIØvelsesgruppe_FK2 FOREIGN KEY (ØvelsesgruppeID)
REFERENCES ØvelsesGruppe (ØvelsesgruppeID)
ON UPDATE CASCADE ON DELETE CASCADE
);

```