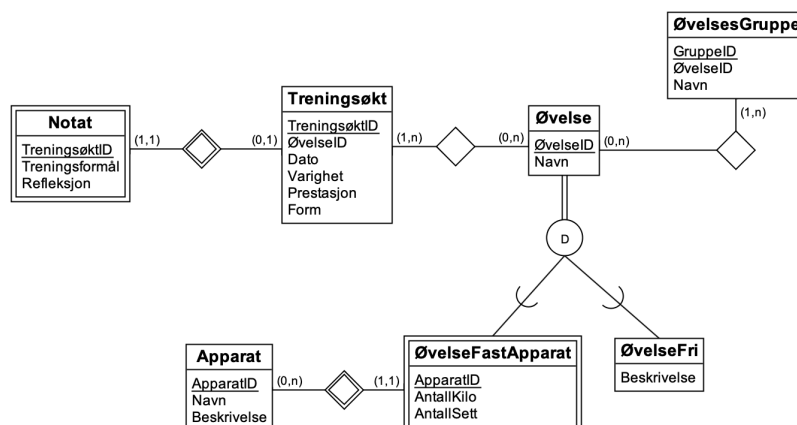


ER-diagram



Figur 1: ER-diagram

Relasjonsdatabasemodell

- **Treningsøkt**(TreningsøktID , Dato, Varighet, Prestasjon, Form)
- **Øvelse**(ØvelseID , Navn)
 - ØvelseID fremmednøkkel til Treningsøkt
- **ØvelseApparat**(ØvelseID , ApparatID, AntallKilo, AntallSett)
 - ØvelseID fremmednøkkel til Treningsøkt, ApparatID fremmednøkkel til Apparat.
- **ØvelseFri**(ØvelseID, Beskrivelse)
 - ØvelseID fremmednøkkel til Treningsøkt.
- **ØvelseØkt**(TreningsøktID, ØvelseID)
 - TreningsøktID fremmednøkkel til Treningsøkt, ØvelseID fremmednøkkel til Øvelse.
- **Notat**(TreningsøktID , Treningsformål, Refleksjon)
 - TreningsøktID fremmednøkkel til Treningsøkt.
- **Apparat**(ApparatID , Navn, Beskrivelse)
- **ØvelsesGruppe**(GruppelID, Navn)
- **ØvelseGruppe**(GruppelID, ØvelseID)
 - GruppelID fremmednøkkel til ØvelsesGruppe, ØvelseID fremmednøkkel til Øvelse.

Beskrivelse av krav

1. Registrere apparater, øvelser og treningsøker med tilhørende data:
 - Et apparat kan registreres uavhengig av alt annet, siden det ikke er en svak klasse.

Dersom øvelsen bruker fast apparat, kan denne referes til, og en øvelse kan dermed opprettes. En treningsøkt kan også opprettes uavhengig. Når både øvelsen og treningsøkten er opprettet, kan id'ene settes inn i `ØvelseIØkt`.

2. Få opp informasjon om et antall n sist gjennomførte treningsøker med notater, der n spesifiseres av brukeren:
 - Siden treningsøktene har en dato, kan man sortere treningsøktene etter dette og velge de n første.
3. For hver enkelt øvelse skal det være mulig å se en resultatlogg i et gitt tidsintervall spesifisert av brukeren:
 -
4. Lage øvelsesgrupper og finne øvelser som er i samme gruppe:
5. Et valgfritt use case som dere selv bestemmer:

SQL-skript