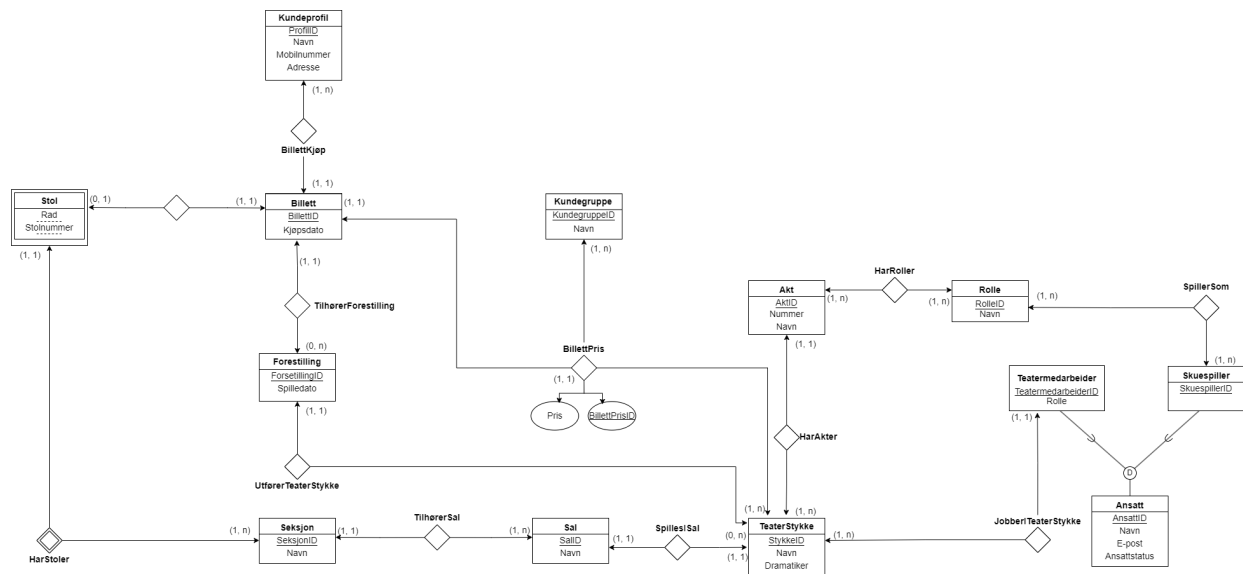


1 ER-Diagram



Figur 1: EER-diagram

For å kunne representere vår miniverden ved hjelp av EER-diagram er vi nødt til å ta noen antagelser. Følgende er våre antagelser.

Ansatt er en entitetsklasse som har attributtene navn, e-post og ansattstatus. Klassen er en moderklasse til to subclasser. Det er en ikke-total disjunkt klasse, siden en ansatt kan enten være en teatermedarbeider, skuespiller eller noe annet. Et eksempel på dette er hvordan teateret har en ansvarlig direktør som jobber ved teateret, men som ikke har en rolle i et teaterstykke.

Teatermedarbeider er en entitetsklasse som har attributtet rolle, og er en subclasse av “Ansatt”. Dette beskriver rollen som personen har i et teaterstykke. En teatermedarbeider vil også ha attributtene til “Ansatt”, og vi antar at hvis en person jobber med et teaterstykke, så jobber personen kun med det teaterstykket, uansett ansattstatus.

Skuespiller er en entitetsklasse som er subclasse av “Ansatt”. En skuespiller vil også ha attributtene til “Ansatt”.

Rolle er en entitetsklasse som har et navn for rollen. “Rolle” og “Skuespiller” har en mange-til-mange relasjon til hverandre siden en rolle kan være spilt av flere skuespillere, og

en skuespiller kan ha flere roller. Siden oppgaven sier at vi ikke trenger å ha oversikt over hvilke forestillinger skuespillere spiller roller, så har vi ikke laget noen direkte relasjoner til forestilling.

Akt er en entitetsklasse som har et nummer og et navn for akten. En akt tilhører et teaterstykke. “Akt” og “Rolle” har en mange-til-mange kobling til hverandre siden en akt kan ha flere roller, og roller kan være i flere akter.

Teaterstykke har et navn og en dramatiker (forfatter). Et teaterstykke vil ha en eller flere teatermedarbeidere. Ut ifra oppgaven og de to teaterstykkene som skal settes inn, er det maks 5 akter, men vi velger fortsatt å sette kardinaliteten til n, siden det kan hende andre teaterstykker har flere akter.

Sal er en entitetsklasse som har et attributt navn. Ut ifra oppgavebeskrivelsen tolker vi det som at et teaterstykke vil kun tilhøre en sal gjennom en sesong. Dermed har vi en en-til-en kobling mellom “Teater” og “Sal”.

Seksjon er en entitetsklasse som har et attributt navn. Siden Hovedscenen (Sal) ikke har noen spesifiserte seksjonsnavn så vil salen kun ha en seksjon som heter parkett. En seksjon vil tilhøre en sal som kan ha flere seksjoner.

Stol er en svak entitetsklasse som har StolNummer og Rad som delvise nøkler. En stol vil være koblet til en seksjon, og en seksjon kan ha flere stoler.

Forestilling er en entitetsklasse som vil vise hvilket teaterstykke som blir spilt til hvilken tid. Siden Spilledato er en dato og tidspunkt vil dette indikere hvilken sesong man er i. Siden våre forestillinger i februar, vet vi at vi er i vinter/vår sesongen. Det kan finnes forestillinger for flere teaterstykker. Men en forestilling kan kun utføre et teaterstykke.

Kundeprofil er en entitetsklasse som vil vise informasjonen til personen som utfører et kjøp av en eller flere billetter. Vi antar her at kunden ikke trenger å lage en brukerkonto, men heller at man fyller ut informasjon ved et kjøp. Dermed vil “Kundeprofil” ha attributtene Navn, Mobilnummer og Adresse.

Kundegruppe er en entitetsklasse som holder oversikt over hvilke kundegrupper man har for billetter. Et eksempel er “barn”. Dermed vil man ha attributtet Navn. Det vil være en mange-til-mange relasjon mellom “Kundegruppe” og “Teaterstykke”. Dette vil utforme “BillettPris” for et teaterstykke. Slik kan de ulike kundegruppene ha ulike priser for ulike teaterstykker.

Billett er en entitetsklasse som viser til en billett som kan bli kjøpt av en kundeprofil. En billett vil tilhøre en stol, en forestilling, en kundeprofil, og ha en billettpris. Hvilken billettpris som blir valgt må bestemmes av logikk i et applikasjonsprogram.

2 Relasjonsdatabaseskjema

Ansatt (AnsattID, Navn, E-post, Ansatt-status)

- AnsattID er primærnøkkel til tabellen

Skuespiller (SkuespillerID)

- SkuespillerID er primærnøkkel i tabellen
- SkuespillerID er fremmednøkkel til “Ansatt”-tabellen. Kan ikke være NULL

Teatermedarbeider (AnsattID, Rolle, *Teaterstykke*)

- AnsattID er fremmednøkkel til “Ansatt”-tabellen. Kan ikke være NULL
- AnsattID er primærnøkkel til tabellen
- Teaterstykke er en fremmednøkkel til “Teaterstykke”-tabellen

Sal (SalID, Navn)

- SalID er primærnøkkel i tabellen

TeaterStykke(StykkeID, Navn, Dramatiker, *Sal*)

- StykkeID er primærnøkkel i tabellen.
- Sal er fremmednøkkel til “Sal”-tabellen. Kan være NULL

Akt (AktID, Nummer, Navn, *TeaterStykke*)

- AktID er primærnøkkel i tabellen.
- TeaterStykke er fremmednøkkel til “TeaterStykke”-tabellen. Kan ikke være NULL

Rolle (RolleID, Navn)

- RolleID er primærnøkkel i tabellen

HarRoller (Akt, Rolle)

- Akt er fremmednøkkel til “Akt”-tabellen. Kan ikke være NULL
- Rolle er fremmednøkkel til “Rolle”-tabellen. Kan ikke være NULL
- Akt og Rolle er primærnøkler i tabellen

SpillerSom (Rolle, Skuespiller):

- Rolle er fremmednøkkel til “Rolle”-tabellen. Kan ikke være NULL
- Skuespiller er fremmednøkkel til “Skuespiller”-tabellen. Kan ikke være NULL
- Rolle og Skuespiller er primærnøkler i tabellen

Seksjon (SeksjonID, Navn, *Sal*)

- SeksjonID er primærnøkkel
- Sal er fremmednøkkel til “Sal”-tabellen. Kan ikke være NULL

Stol (Rad, StolNummer, Seksjon)

- Rad, StolNummer, Seksjon er primærnøkkel
- Seksjon er fremmednøkkel til Seksjon-tabellen

Forestilling (ForstillingID, Spilledato, *TeaterStykke*)

- ForestillingID er primærnøkkel i tabellen
- Teaterstykke er fremmednøkkel til “TeaterStykke”-tabellen. Kan ikke være NULL
- (Spilledato og TeaterStykke) er en supernøkkel i tabellen. Spilledato er av typen DateTime, så det vil ikke være mulig å ha to forestillinger med samme teaterstykke til samme tid, siden oppgaven tilsier at et teaterstykke spilles i kun en sal.

Kundeprofil (ProfilID, Navn, Mobilnummer, Adresse)

- ProfilID er primærnøkkel

Kundegruppe (KundegruppeID, Navn)

- KundegruppeID er primærnøkkel til tabellen.

BillettPris (BillettPrisID, Pris, *Kundegruppe*, *TeaterStykke*)

- BillettPrisID er primærnøkkel til tabellen. Dette er primærnøkkelen siden det er lettere for andre tabeller (f.eks. "Billett") å koble til tabellen med kun en nøkkel istedenfor to.
- Kundegruppe er fremmednøkkel til "Kundegruppe"-tabellen. Kan ikke være NULL
- TeaterStykke er fremmednøkkel til "TeaterStykke"-tabellen. Kan ikke være NULL
- Kundegruppe og TeaterStykke er en supernøkkel sammen siden det ikke kan være samme kundegruppe til samme teaterstykke flere ganger.

Billett (BillettID, Type, Kjøpsdato, *Rad*, *StolNummer*, *Seksjon*, *Forestilling*, *Kundeprofil*, *BillettPris*)

- BillettID er primærnøkkel i tabellen
- Rad, StolNummer og SeksjonID er fremmednøkler til "Stol"-tabellen. De kan ikke være NULL.
- ForestillingID er fremmednøkkel til "Forestilling"-tabellen. Kan ikke være NULL.
- KundeprofilID er fremmednøkkel til "Kundeprofil"-tabellen. Kan ikke være NULL
- BillettPris er fremmednøkkel til "BillettPris"-tabellen.
- Rad, StolNummer, Seksjon og Forestilling er sammen en supernøkkel. Dette kommer av at en forestilling ikke kan ha flere billetter på samme sete.

2.1 - 4NF

- En tabell er på 4. Normalform hvis der for alle ikke-trivielle MVD-er, $X \twoheadrightarrow Y$, er slik at X er en supernøkkel. Det vil at også si at en tabell på 4NF også vil være på BCNF.
- BCNF er oppfylt når det for alle funksjonelle avhengigheter er slik at venstresiden er en supernøkkel i tabellen.

- De fleste tabellene i vår oversikt er svært enkle, med kun primærnøkkel på venstresiden.
- Ingen av våre tabeller har MVD-er, altså $X \twoheadrightarrow Y$, kan være en mengde Y-verdier som er assosiert med X-verdien.
- I for eksempel Ansatt tabellen og TeaterStykke tabellen har vi primærnøkler som er en unik identifikator for hver rad i tabellen. Altså ingen MVD-er.
- HarRolle er en mange-til-mange relasjonstabell som kobler to entiteter sammen. Her er det de to fremmednøklerne som sammen danner primærnøkkelen.
- Vi har tabeller der hvor attributter er funksjonelt avhengig av attributter som ikke inngår i primærnøkkelen for tabellen, men i disse tabellene er venstresiden supernøkkel i tabellen, slik at vi har BCNF. For eksempel i Forestilling-tabellen, har vi forestillingen som er funksjonelt avhengig av Spilledato og Teaterstykke.

3 SQL

Vi har opprettet et SQL script som lager tabeller og fører inn i en database. Vi har fylt inn CONSTRAINTs for alle fremmednøkler selv, siden det er “best practice” å gjøre det siden man da får lett oversikt over navnene. På denne måten er det enkelt å endre en CONSTRAINT hvis det er nødvendig.

Vi har også benyttet oss av en CONSTRAINT UNIQUE på “Forestilling” og “BillettPris”. For “Forestilling” kan det ikke være samme teaterstykke til samme spilledato flere ganger, slik som forklart i oppgave 2. For “BillettPris” må en sammenkobling av kundegruppe og teaterstykke være unik. Det gir ikke mening at det skal være duplikat av kundegruppe mot et teaterstykke.

Videre har vi satt et CONSTRAINT på prisen til “BillettPris” for å sjekke at prisen er over 0 NOK.

Vi vurderte å legge til en CHECK CONSTRAINT på f.eks. Navn på kundegrupper og seksjonsnavn. Men vi konkluderte med at dette er tungvint hvis man ønsker å legge til en ny

“Kundegruppe” eller “Seksjon” som ikke er nevnt i oppgaven. Det samme gjelder for prisene for en “BillettPris”.

Til slutt har vi benyttet oss av ON DELETE CASCADE på alle fremmednøkler bortsett fra “TeaterStykke” sin fremmednøkkel til “Sal”. Dette kommer av at hvis en sal slettes så vil vi ikke at teaterstykket og all data koblet til dette skal slettes automatisk. Da vil vi heller at man kan tilegne teaterstykket en ny sal. Dermed kan fremmednøkkelen være NULL. Det kan ikke de resterende fremmednøkler. Sal er også satt til å være UNIQUE siden det ikke skal kunne spilles flere teaterstykker i en Sal, siden det skal bli lettere med kulisser og utstyr.

Som nevnt i oppgave 2, så kan vi ikke sette inn en CONSTRAINT for hvilken “BillettPris” som blir koblet til en “Billett”. Dette må skje via logikk i et applikasjonsprogram.

3.1 Instruksjoner for utførelse av SQL:

- Åpne mappen i en terminal, slik at du har create.sql filen i mappen.
- For å opprette en database, skriv: `sqlite3 teater.db`
- For å fylle databasen med tabellene, skriv: `.read create.sql`
- For å sjekke at tabellene eksisterer, skriv: `.tables`