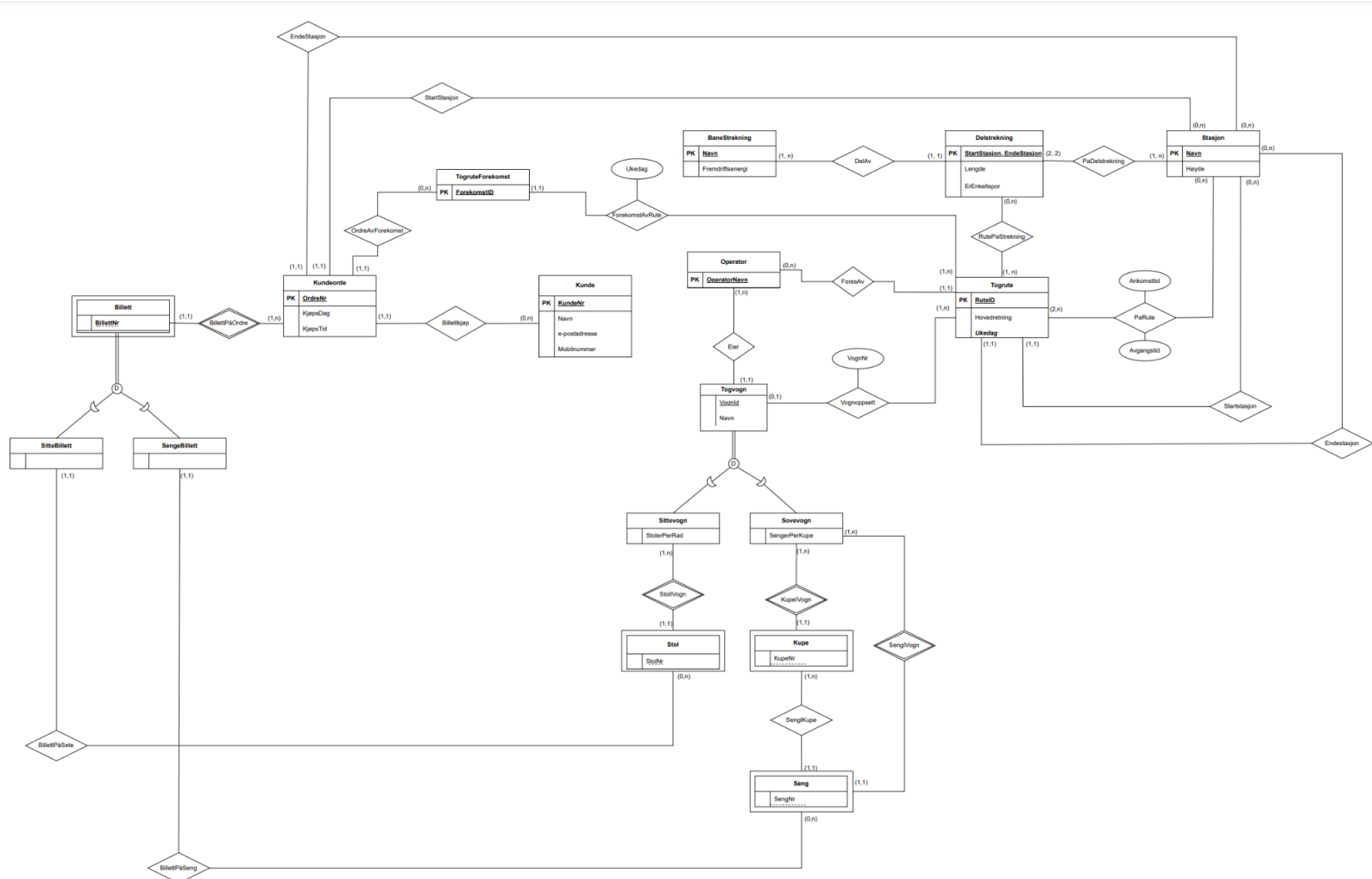


# Prosjekt TDT4145 - DB1

ER-modell:



*Mer oversiktlig versjon av diagrammet er lagt ved som egen fil*

### Forutsetninger og valg:

- Ankomst- og avgangstiden i relasjonen mellom rute og stasjon kan være null for å ta hensyn til at det kan være en start- eller endestasjon
- En delstrekning kan ikke være del av flere banestrekninger
- En stasjon kan være del av flere delstrekninger (f.eks Oslo S som har jernbaner som går flere enn to veier)
- Delstrekninger har start- og slutt-stasjoner hvor de sammen bestemmer start- og slutt-stasjon for hele banestrekningen
- En vogn kan ikke eies av flere operatører
- I stedet for “vogntype” har vi “vogn” klassen hvor vi innfører en unik vognID
- Alle kupeer i en sovevogn inneholder like mange senger, samme gjelder seter per rad i sittevogn

- En togrute må ha minst en passasjervogn
- En togruteforekomst må ha informasjon om ukedagen ruten kjøres på, slik at dag kan knyttes til kundeordren
- Alle billetter i en kundeordre kjøres på samme strekning
- Alle operatører har et felles kunderegister som er mengden av alle kunde entiteter
- Sengnummer velges uavhengig av kupe, men må fortsatt være relatert til kupe pga. restriksjoner av billettkjøp. Derfor er sengens identifiserende relasjon til sovevogn, med en vanlig relasjon til kupe.
- Innførte surrogatnøkler:
  - VognID
  - RutelID
  - ForekomstID
- Unike attributter:
  - Antar at navnet til en banestrekning er unikt
  - Antar at navnet til en operatør er unikt
  - Antar at stasjoner har unike navn

### Merknader:

- Ukedag-attributtet i Togrute er en MVD. Dette blir tatt hensyn for i relasjonsdatabaseskjemaet for å oppnå 4NF.
- ErEnkeltspor i Delstrekning, Hovedretning i Togrute og Type i Vogn er Booleans, men dette støttes ikke i SQL-lite. Dette må derfor tas hensyn til med 1 og 0 verdier, altså INTEGER, for true og false.

### Relasjonsdatabaseskjema:

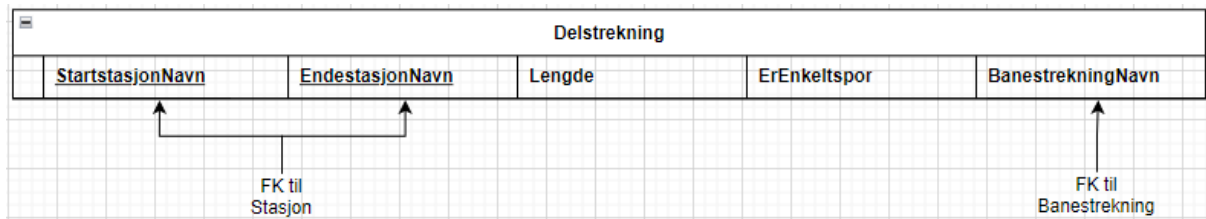
I utgangspunktet er alle tabellene på 4NF, hvor attributt(ene) som er understreket definerer primærnøkkel.

Banestrekning				
<u>Navn</u>				Fremdriftsenergi

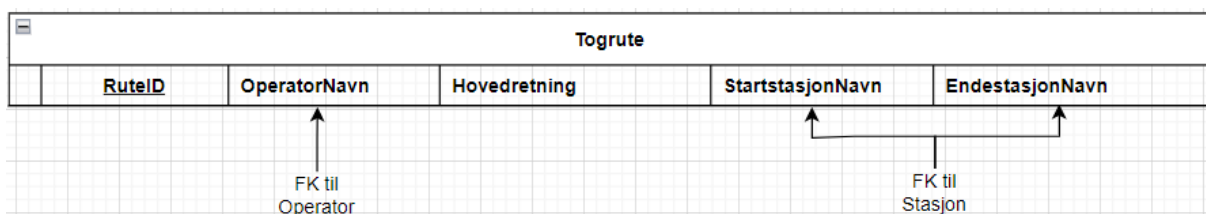
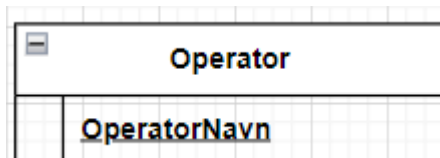
Navn → Fremdriftsenergi

Stasjon				
<u>Navn</u>				Høyde

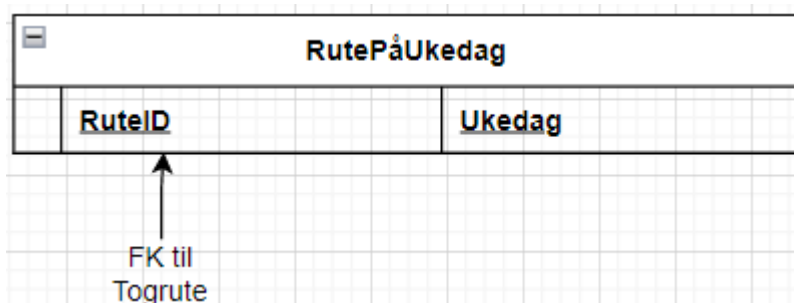
Navn → Høyde



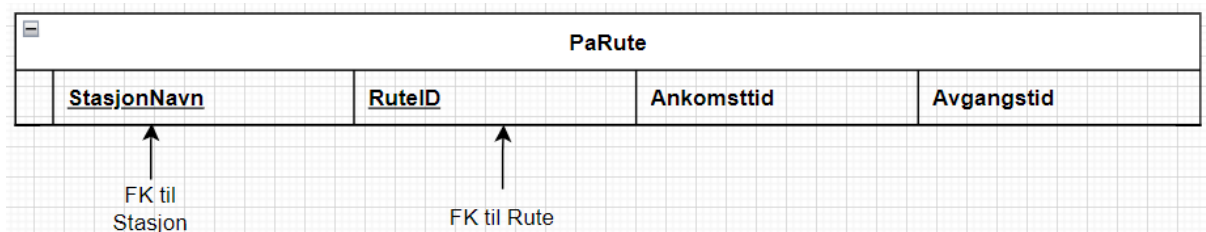
StartstasjonNavn, EndestasjonNavn → Lengde, ErEnkeltSpor, BaneStrekningNavn  
 Her antar vi at en delstrekning bare kan være del av en banestrekning. Dette gir fortsatt grunnlag for at en stasjon kan inngå i flere banestrekninger.



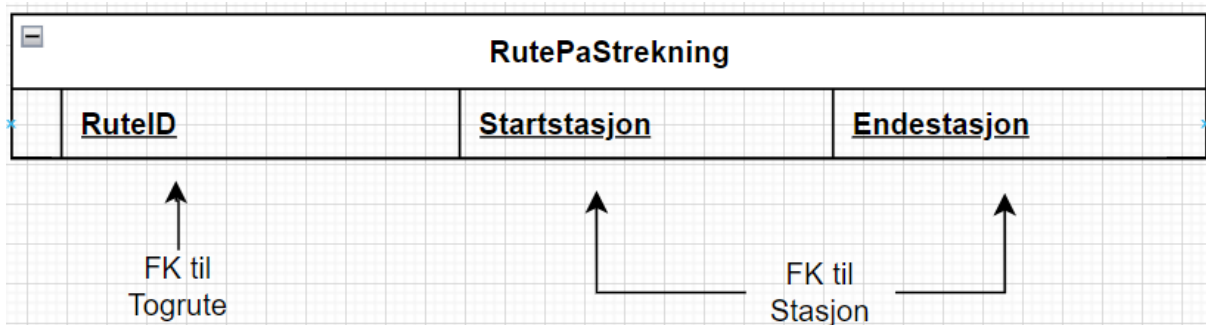
RuteID → OperatorNavn, Hovedretning, StartstasjonNavn, EndestasjonNavn  
 Start- og ende-stasjon må sammen med hovedretning samsvare med retningen på delstrekning(ene) som ruten kjøres på, noe som håndteres i applikasjonen.



For å unngå MVD, innførte vi en egen tabell for å holde oversikt på hvilke ukedager en rute kjøres. Denne løsningen fører til at Togrute og RutePåUkedag er på 4NF.

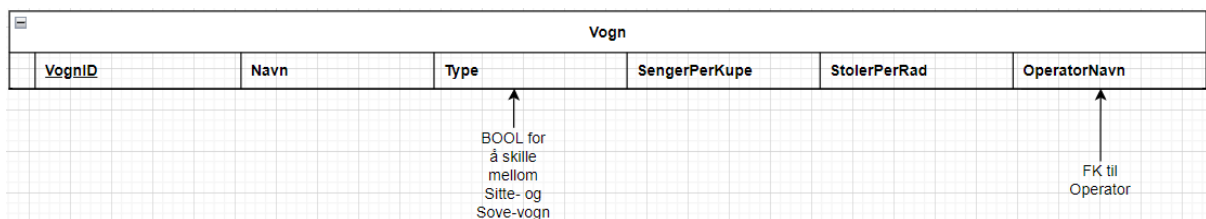


StasjonNavn, RuteID → Ankomsttid, Avgangstid  
 Relasjonstabell mellom Rute og Stasjon. Mengden PaRute-entiteter med samme RuteID tilsvarer "Togrutetabellen" som er spesifisert i oppgaven.



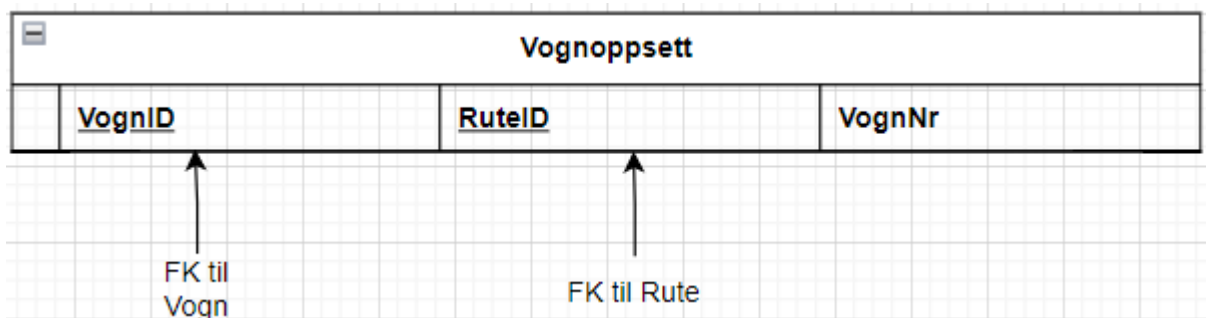
Relasjonstabell mellom Rute og Delstrekning.

Mengden RutePaStrekning entiteter fra samme RuteID beskriver ruten. Start- og Ende-Stasjon må samsvare med rutens retning, noe som håndteres i applikasjonen.



VognID → Navn, Type,engerPerKupe, StolerPerRad, OperatorNavn

Vi valgte å løse spesialiseringen av togvogner ved å kombinere dem i en felles tabell å innføre type-attributt. Her vil NULL-verdier oppstå iengerPerKupe og StolerPerRad. Vi valgte denne løsningen for å enklere kunne knytte superklassen Togvogn mot en rute sitt Vognoppsett. Ett vognoppsett må ha en VognID som fremmednøkkel, noe som hadde vært vanskelig å løse dersom vogntypene var delt inn i to tabeller.



VognID, RuteID → VognNr

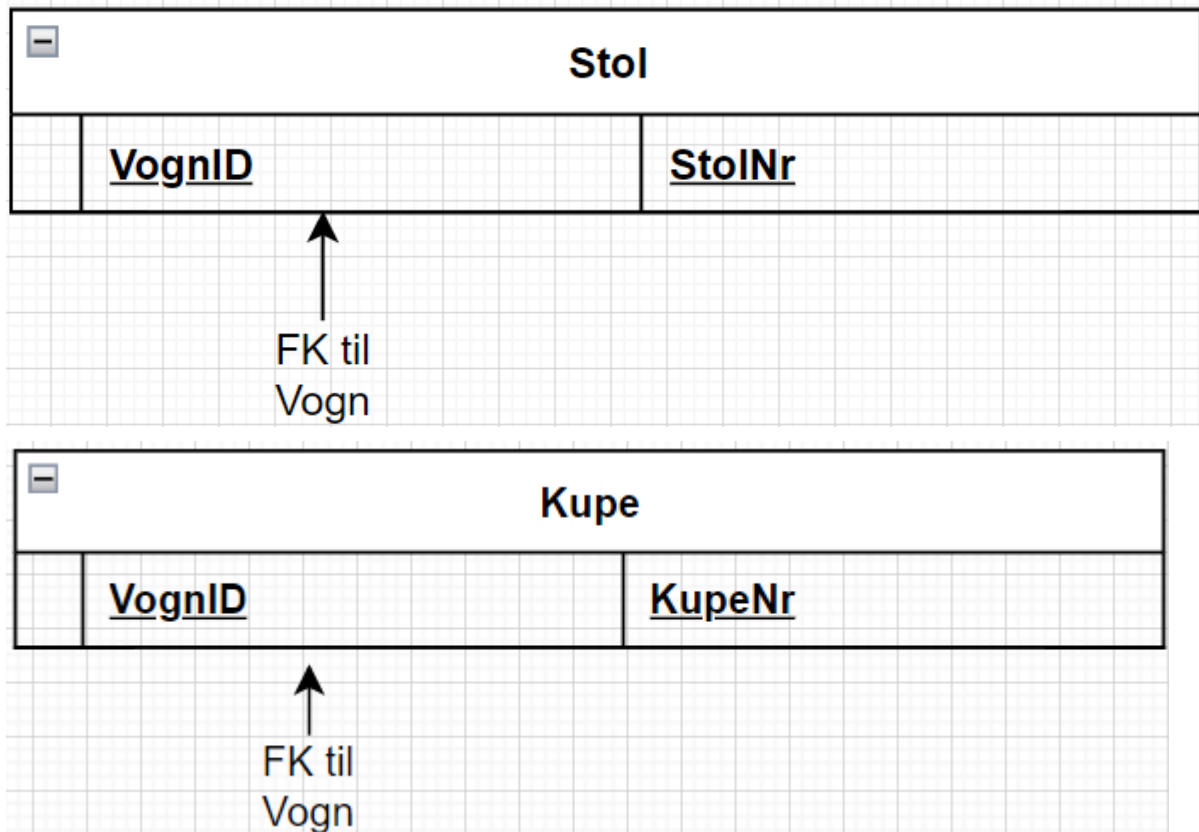
En vogn kan dermed i teorien være en del av flere ruter dersom rutene ikke kjøres samtidig. Samme vogn kan da ha forskjellig nummer ut ifra hvilken rute den inngår i. Applikasjonen må ta hensyn til at en vogn ikke blir brukt flere steder samtidig. Vi valgte denne løsningen på grunn av oppgavens beskrivelse av vognnummerering i et vognoppsett.

Vi har i tillegg den funksjonelle avhengigheten:

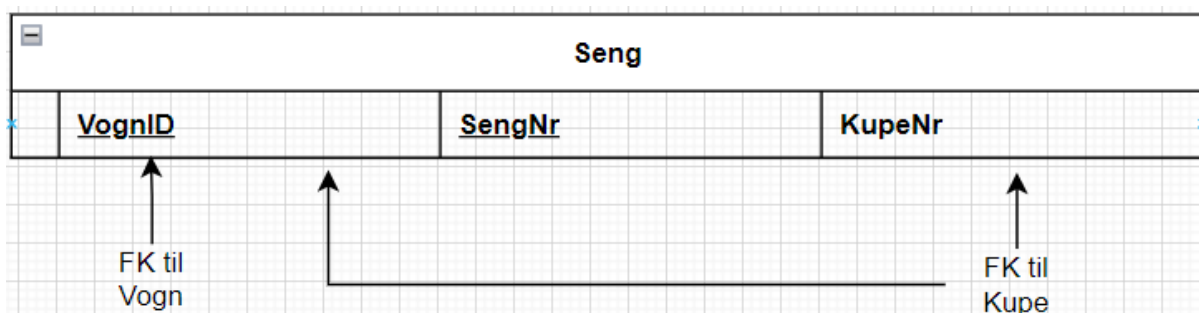
RuteID, VognNr → VognID

Altså på grunn av oppgavens beskrivelse av vognnummerering i et vognoppsett, finnes det bare en Vogn i en Rute med samme VognNr. Applikasjonen tar hensyn til dette.

Til tross for denne funksjonelle avhengigheten, er tabellen på 4NF siden VognID er et nøkkelattributt og (RuteID, VognNr) er en supernøkkel.

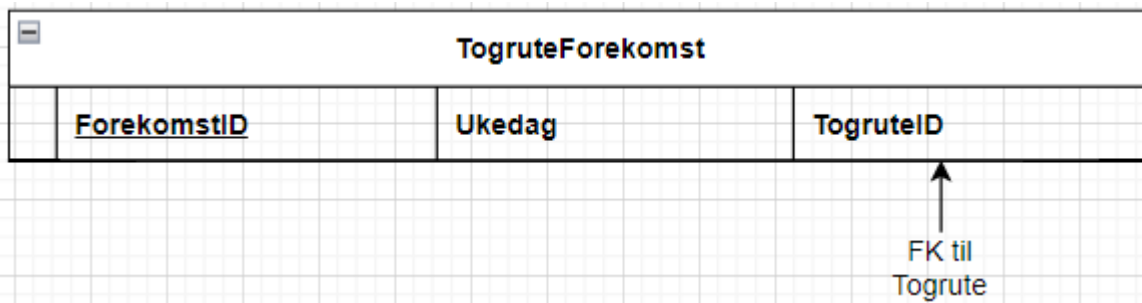


Applikasjonen må sjekke at Stolen/Kupeen er relatert til en Vogn av Type = Sittevogn/Sovevogn.



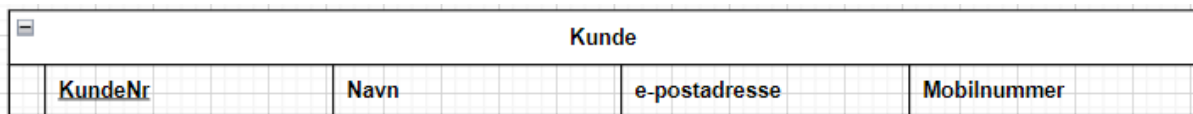
VognID, SengNr → KupeNr

Nummerering av Senger er uavhengig av Kupe, så derfor har vi valgt å bruke Vogn til å identifisere Seng-entiteter. På grunn av restriksjoner under bestilling av Senger i Kupeer, må Seng fortsatt ha en relasjon til Kupe. En Seng sin plassering i Kupeen må beregnes i applikasjonen.

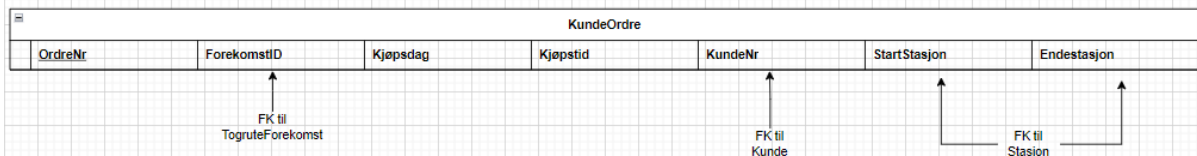


ForekomstID → Ukedag, TogruteID

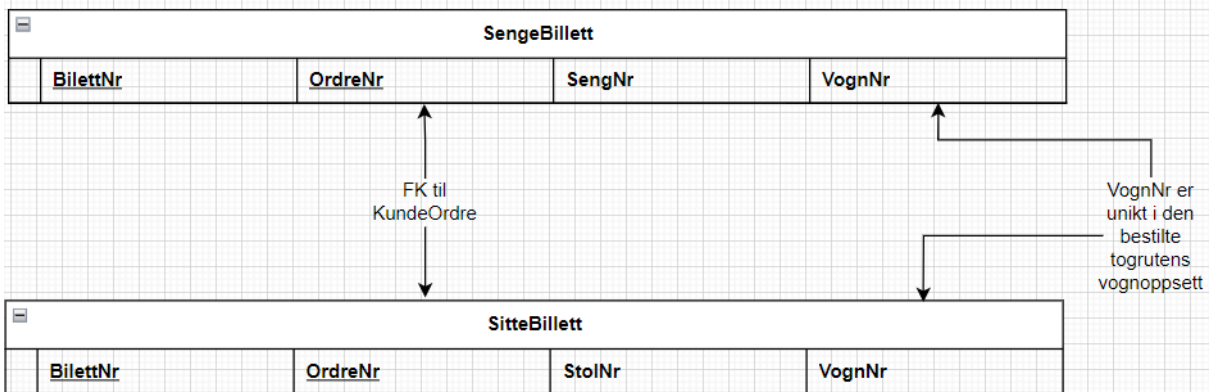
TogruteForekomst-entiteter må være på en Ukedag som Togruten kjøres på. Dette må applikasjonen sørge for.



KundeNr → Navn, e-postadresse, Mobilnummer



OrdreNr → ForekomstID, Kjøpsdag, Kjøpstid, KundeNr, StartStasjon, Endestasjon



BillettNr, OrdreNr → SengNr/StolNr, VognNr

Her har vi valgt å knytte Billetter til VognNr istedet for VognID, slik at Kunden får informasjon om hvilken vogn hen skal gå på. VognNr sammen med OrdreNr kan identifisere hvilken VognID og StolNr billetten tilhører.

OrdreNr → ForekomstID → RutelID

RutelID, VognNr → VognID

Dermed bevarer vi referanseintegritet og kan identifisere Stol-entiteten

