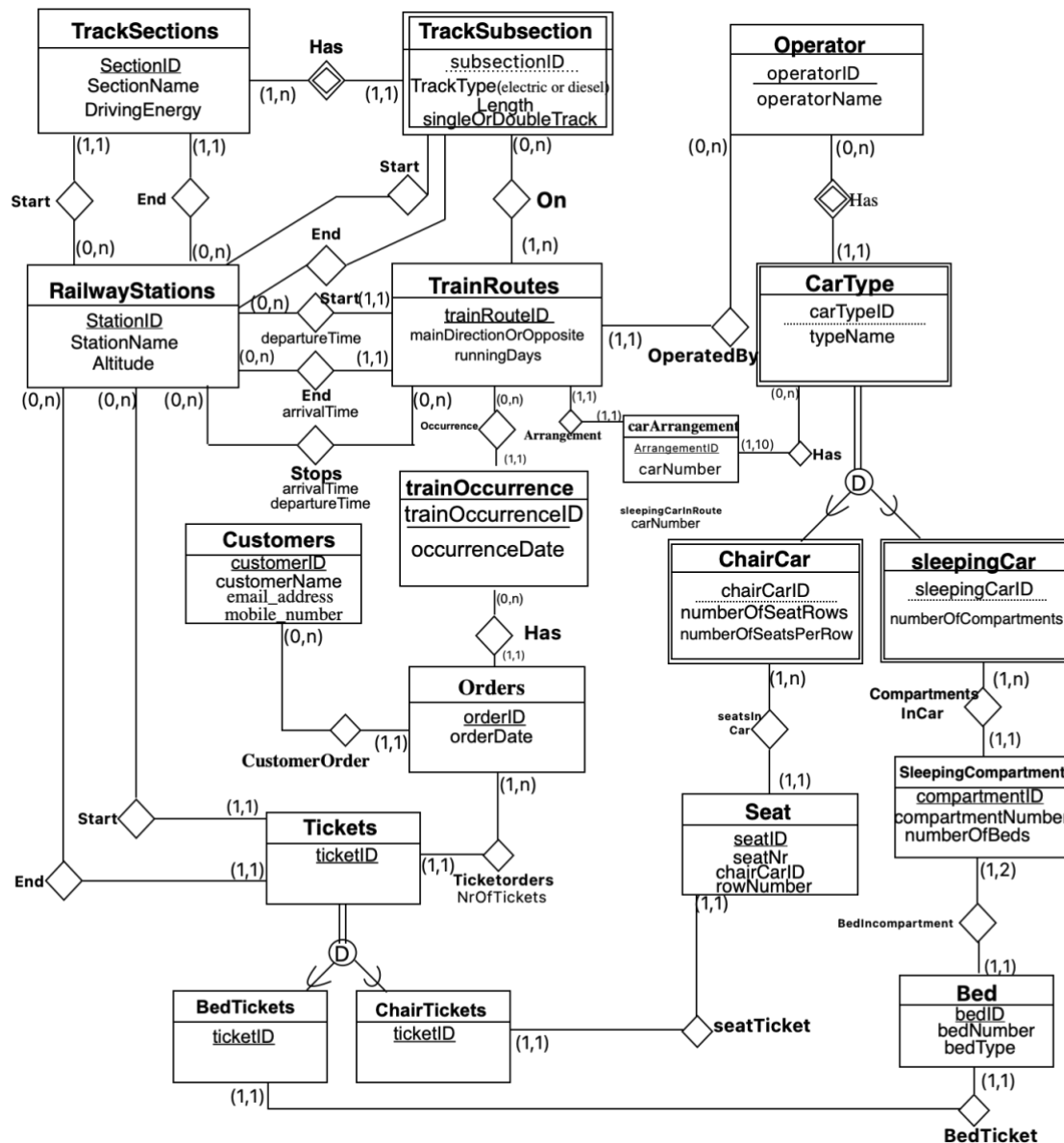


Navn- Kebene Amanuel Baissa

Innlevering, del 1

A) ER MODELL



Antagelsene jeg har gjort kommer med relasjonsdatabaseskjema (tabeller under).

B) Relasjonsdatabaseskjema

RailwayStations

<u>StationID</u>	StationName	Altitude
------------------	-------------	----------

TrackSection

<u>SectionID</u>	SectionName	DrivingEnergy	StartStationID(FK)	EndStationID(FK)
------------------	-------------	---------------	--------------------	------------------

SectionID er fremmednøkkel mot Tracksections og er primærnøkkelen Og kan ikke være NULL. DrivingEnergy kan være diesel eller electric. Her kan jeg bruke BOOLEAN også men ikke alle sql systemer støtter denne funksjonen. StartStationID og EndStationID fremmednøkkeler mot RailwayStation. Jeg antar at det ikke finnes flere banestrekninger mellom to like stasjoner i Norge og alle har forskjellige ID

Subsection

<u>SubsectionID</u>	<u>SectionID(FK)</u>	TrackType	Length	trackType	StartStationID(FK)	EndStationID(FK)
---------------------	----------------------	-----------	--------	-----------	--------------------	------------------

Er en svak klasse av TrackSection og har en delvis nøkkel SubsectionID. SubsectionID er også fremmednøkkel i TrackSubsection. SubsectionID sammen med SectionID er primærnøkkelen. StartStationID og EndStationID er fremmednøkkeler mot RailwayStations og kan ikke være NULL.

Operator

<u>operatorID</u>	operatorName
-------------------	--------------

operatorID er primærnøkkelen.

CarType

<u>carTypeID</u>	<u>operatorID(FK)</u>	typeName
------------------	-----------------------	----------

CarType er en svak klasse av Operator og carTypeID er delvis nøkkel i denne tabbelen og operatorID er fremmednøkkel mot Operator. Og sammen er de primærnøkkel til CarType. Denne tabellen inneholder navn og Id sammen. Vi har to CarType. ChairCar og SleepingCar. Dette er en disjunkt og total spesialisering med CarType som superklasse og ChairCar og SleepingCar som subklasser.

ChairCar

<u>chairCarID</u>	<u>operatorID(FK)</u>	CarTypeID(FK)	numberOfSeatRows	numberOfSeatsPerRow
-------------------	-----------------------	---------------	------------------	---------------------

ChairCar er subklasse av svak klassen av CarType og dermed svak klasse med delvisnøkkel chairCarID og primærnøkkelen er chaircarID og operatorID sammen. operatorID er fremmednøkkel mot Operator og CarTypeID er fremmednøkkel mot CarType.

SleepingCar

<u>sleepingCarID</u>	<u>operatorID(FK)</u>	CarTypeID(FK)	numberOfCompartments
----------------------	-----------------------	---------------	----------------------

Er subklasse av svak klassen av CarType og dermed svak klasse med delvisnøkkel chairCarID og primærnøkkelen er chaircarID og operatorID sammen. operatorID er fremmednøkkel mot Operator og CarTypeID er fremmednøkkel mot CarType.

Seat

<u>seatID</u>	seatNr	rowNumber	chairCarID(FK)	<u>operatorID(FK)</u>
---------------	--------	-----------	----------------	-----------------------

SeatID er primærnøkkelen og chairCarID og operatorID er fremmednøkkel mot ChairCar.

SleepingCompartment

<u>compartmentID</u>	compartmentNumber	numberOfBeds	sleepingCarID(FK)	operatorID(FK)
----------------------	-------------------	--------------	-------------------	----------------

compartmentID er primærnøkkelen. sleepingCarID og operatorID er fremmednøkkel mot SleepingCar og kan ikke være NULL.

Bed

<u>bedID</u>	compartmentID(FK)	sleepingCarID(FK)	bedNumber	bedType
--------------	-------------------	-------------------	-----------	---------

compartmentID og sleepingCarID er fremmednøkkel mot SleepingCompartment og kan ikke være NULL.

CarArrangements

<u>arrangementID</u>	carNumber	carTypeID(FK)	operatorID(FK)
----------------------	-----------	---------------	----------------

CarTypeID og operatorID er fremmednøkkel mot CarType og kan ikke være NULL.

StartStation

<u>startStationId(FK)</u>	departureTime
---------------------------	---------------

startStationID er fremmednøkkel mot RailwayStations. Denne tabellen ble splittet fra TrainRoutes for å ikke ha redundans og ha den på 4NF.

EndStation

<u>endStationId(FK)</u>	arrivalTime
-------------------------	-------------

endStationID er fremmednøkkel mot RailwayStations. Denne tabellen ble splittet fra TrainRoutes for å ikke ha redundans.

TrainRoutes

<u>trainRouteID</u>	OperatorID(FK)	startStationId(FK)	endStationId(FK)	Direction (mainOrNot)	runningDays
---------------------	----------------	--------------------	------------------	--------------------------	-------------

OperatorId er fremmednøkkel mot Operator og jeg antar at en rute blir kjørt av bare en Operator. startStationId og endStationId er fremmednøkkel mot StartStation og StopStation henholdsvis. runningDays tar inn CHAR(7). For eksempel 'SMTWTFS' for alle dager i uken.

TrainStops

<u>trainStopId(FK)</u>	trainRouteId(FK)	arrivalTime	departureTime
------------------------	------------------	-------------	---------------

trainStopID er fremmednøkkel mot RailwayStations og er primærnøkkel for TrainStops tabellen. trainRouteID er fremmednøkkel mot TrainRoutes. Her får vi en tabell på alle stoppene til en rute med avgang og ankomst. (uten start og slutt stasjon)

RouteCarArrangements

<u>trainRouteID(FK)</u>	arrangementID(FK)
-------------------------	-------------------

trainRouteID er fremmednøkkel mot TrainRoutes og arrangementID er fremmednøkkel mot CarArrangements og de kan ikke være NULL.

TrainOccurrence

<u>occurrenceID</u>	trainRouteID(FK)	occurrenceDate
---------------------	------------------	----------------

trainRouteID er fremmednøkkel mot TrainRoutes og kan ikke være NULL.

Customers

<u>customerID</u>	customerName	email	mobile
-------------------	--------------	-------	--------

CustomerOrders

<u>orderID</u>	customerID(FK)	occurrenceID(FK)	orderDate	numChairTickets	numBedTickets
----------------	----------------	------------------	-----------	-----------------	---------------

customerID er fremmednøkkel mot Customers og occurrenceID er fremmednøkkel mot TrainOccurrence.

ChairTickets

<u>ticketID</u>	orderID(FK)	seatID(FK)	startStationId(FK)	EndStationId(FK)	OccUrrrenceID
-----------------	-------------	------------	--------------------	------------------	---------------

StartStationID og EndStationID er fremmednøkkel mot RailwayStations og kan ikke være NULL.

OrderID er fremmednøkkel mot CustomerOrders. seatID er fremmednøkkel mot Seat og occurrenceID er fremmednøkkel mot TrainOccurence.

Bedtickets

<u>ticketID</u>	orderID(FK)	bedID(FK)	startStationId(FK)	endStationId(FK)
-----------------	-------------	-----------	--------------------	------------------

StartStationID og EndStationID er fremmednøkkel mot RailwayStations og kan ikke være NULL. OrderID er fremmednøkkel mot CustomerOrders. occurrenceID er fremmednøkkel mot TrainOccurence. bedID er fremmednøkkel mot bed.

Alle tabeller er på 4NF fordi det ikke finnes FA-er eller MVD-er i tabellene og gir grunnlag for uønsket redundans.

Tabellene representerer i stor grad ER modellen i A. Og oppsplittene jeg har gjort har jeg kommentert under relasjonsdatabaseskjema tabellene.

Unntak er TrackSection og SubSection som bryter BCNF for at det er FD mellom kandidatnøkler. Men siden det ikke fører til uønsket redundans og de blir oversiktlige tabeller har jeg valgt å gjøre det på denne måten. Det gjør også at vi slipper å legge flere tabeller sammen.

Til de fleste tabellene har jeg innført surrogat nøkkel for at nøkkelen blir sterkere og under kontroll. Fremmednøkkel i en tabell har jeg merket med (FK) primær nøkkel i grunn rute. Nøkler er mer spesifisert i SQL script i C.

C) SQL script som konstruerer databasen med tabeller

```
CREATE TABLE RailwayStations (  
    stationID    INTEGER NOT NULL,  
    stationName VARCHAR(100),  
    altitude    FLOAT,  
    PRIMARY KEY(stationID)  
);
```

```
CREATE TABLE TrackSection (  
    sectionID    INTEGER NOT NULL PRIMARY KEY,  
    sectionName  VARCHAR(100) NOT NULL,  
    drivingEnergy VARCHAR(100) NOT NULL,  
    startStationID INT NOT NULL,  
    endStationID INT NOT NULL,  
    FOREIGN KEY (startStationID) REFERENCES RailwayStations(stationID),  
    FOREIGN KEY (endStationID) REFERENCES RailwayStations(stationID)  
);
```

```

CREATE TABLE SubSections (
    subSectionID INTEGER NOT NULL,
    trackSectionID INTEGER NOT NULL,
    subSectionLength_km FLOAT,
    trackType VARCHAR(10),
    startStationId INTEGER NOT NULL,
    endStationID INTEGER NOT NULL,
    PRIMARY KEY(subSectionID,trackSectionID)
    FOREIGN KEY (trackSectionID) REFERENCES TrackSection(sectionID),
    FOREIGN KEY (startStationId) REFERENCES RailwayStations(stationID),
    FOREIGN KEY (endStationID) REFERENCES RailwayStations(stationID)
);

```

```

CREATE TABLE Operator (
    operatorID INTEGER NOT NULL PRIMARY KEY,
    operatorName VARCHAR(100)
);

```

```

CREATE TABLE CarType (
    carTypeID INTEGER NOT NULL,
    operatorID INTEGER NOT NULL,
    typeName VARCHAR(100),
    PRIMARY KEY ( carTypeID, operatorID),
    FOREIGN KEY (operatorID) REFERENCES Operator(operatorID)
    ON DELETE CASCADE
);

```

```

CREATE TABLE ChairCar (
    chairCarID INTEGER NOT NULL,
    CarTypeID INTEGER NOT NULL,
    operatorID INTEGER NOT NULL,
    numberOfSeatRows INTEGER,
    numberOfSeatsPerRow INTEGER,
    PRIMARY KEY (chairCarID,operatorID),
    FOREIGN KEY (CarTypeID,operatorID) REFERENCES
    CarType(carTypeID,operatorID)
);

```

```

CREATE TABLE SleepingCar (

```

```
sleepingCarID INTEGER NOT NULL,  
carTypeID INTEGER NOT NULL,  
operatorID INTEGER NOT NULL,  
numberOfCompartments INTEGER,  
PRIMARY KEY (sleepingCarID,operatorID),  
FOREIGN KEY (carTypeID,operatorID) REFERENCES  
CarType(carTypeID,operatorID)  
);
```

```
CREATE TABLE SleepingCompartment (  
compartmentID INTEGER NOT NULL,  
sleepingCarID INTEGER NOT NULL,  
compartmentNumber INTEGER,  
numberOfBeds INTEGER,  
operatorID INTEGER NOT NULL,  
PRIMARY KEY (compartmentID),  
FOREIGN KEY (sleepingCarID, operatorID) REFERENCES  
SleepingCar(sleepingCarID, operatorID)  
);
```

```
CREATE TABLE Bed (  
bedID INTEGER NOT NULL PRIMARY KEY,  
compartmentID INTEGER NOT NULL,  
bedNumber INTEGER NOT NULL,  
bedType VARCHAR(10),  
FOREIGN KEY (compartmentID) REFERENCES  
SleepingCompartment(compartmentID)  
);
```

```
CREATE TABLE Seat (  
seatID INTEGER PRIMARY KEY,  
seatNr INTEGER NOT NULL,  
rowNumber INTEGER NOT NULL,  
chairCarID INTEGER NOT NULL,  
operatorID INTEGER NOT NULL,  
FOREIGN KEY (chairCarID,operatorID) REFERENCES  
ChairCar(chairCarID,operatorID)  
);
```

```
CREATE TABLE CarArrangements (  
arrangementID INTEGER NOT NULL PRIMARY KEY,
```

```

carNumber INTEGER NOT NULL,
carTypeID INTEGER NOT NULL,
operatorID INTEGER NOT NULL,
FOREIGN KEY (carTypeID, operatorID) REFERENCES CarType(carTypeID,
operatorID)
);
CREATE TABLE StartStation(
startStationId INTEGER NOT NULL PRIMARY KEY,
departureTime TIME NOT NULL,
FOREIGN KEY (startStationId) REFERENCES RailwayStations(stationID)
);

CREATE TABLE EndStation(
endStationId INTEGER NOT NULL PRIMARY KEY,
arrivalTime TIME NOT NULL,
FOREIGN KEY (endStationId) REFERENCES RailwayStations(stationID)
);
CREATE TABLE TrainRoutes (
trainRouteId INTEGER NOT NULL PRIMARY KEY,
operatorID INTEGER NOT NULL,
startStationId INTEGER NOT NULL,
endStationId INTEGER NOT NULL,
direction VARCHAR(10) NOT NULL,
runningDays CHAR(7) NOT NULL,
FOREIGN KEY (startStationId) REFERENCES StartStation(stationID),
FOREIGN KEY (endStationId) REFERENCES EndStation(stationID),
FOREIGN KEY (operatorID) REFERENCES Operator(operatorID)
);

CREATE TABLE TrainStops (
trainStopId INT PRIMARY KEY,
trainRouteId INT NOT NULL,
arrivalTime TIME NOT NULL,
departureTime TIME NOT NULL,
FOREIGN KEY (trainRouteId) REFERENCES TrainRoutes(trainRouteId),
FOREIGN KEY (trainStopId) REFERENCES RailwayStations(stationID)
);

CREATE TABLE RouteCarArrangements(
trainRouteId INTEGER NOT NULL PRIMARY KEY,

```



```
arrangementID INTEGER NOT NULL,  
  FOREIGN KEY (trainRouteID) REFERENCES TrainRoutes(trainRouteID) ,  
  FOREIGN KEY (arrangementID) REFERENCES CarArrangements(arrangementID)  
);
```

```
CREATE TABLE TrainOccurrence (  
  occurrenceID INTEGER NOT NULL PRIMARY KEY,  
  trainRouteID INTEGER NOT NULL,  
  occurrenceDate DATE NOT NULL,  
    FOREIGN KEY (trainRouteID) REFERENCES TrainRoutes(trainRouteID)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE Customers (  
  customerID INTEGER PRIMARY KEY,  
  customerName VARCHAR (100),  
  email VARCHAR (100),  
  mobile VARCHAR(10)  
);
```

```
CREATE TABLE CustomerOrders (  
  orderID INTEGER PRIMARY KEY,  
  customerID INTEGER NOT NULL,  
  occurrenceID INTEGER,  
  orderDate DATE NOT NULL,  
  numChairTickets INTEGER,  
  numBedTickets INTEGER  
);
```

```
CREATE TABLE ChairTickets (  
  ticketID INTEGER PRIMARY KEY,  
  occurrenceID INTEGER NOT NULL,  
  orderID INTEGER NOT NULL,  
  seatID INTEGER NOT NULL,  
  startStationId INTEGER NOT NULL,  
  endStationId INTEGER NOT NULL,  
  FOREIGN KEY(orderID) REFERENCES CustomerOrders(orderID),  
  FOREIGN KEY (startStationId) REFERENCES RailwayStations(stationID),  
  FOREIGN KEY (endStationID) REFERENCES RailwayStations(stationID),  
  FOREIGN KEY(seatID) REFERENCES Seat(seatID),
```

```

FOREIGN KEY(occurrenceID) REFERENCES TrainOccurrence(occurrenceID),
CONSTRAINT uniqueTicket UNIQUE (occurrenceID, seatID, startStationId,
endStationId)
);

```

```

CREATE TABLE Bedtickets (
ticketID INTEGER PRIMARY KEY,
orderID INTEGER NOT NULL,
occurrenceID INTEGER NOT NULL,
bedID INTEGER NOT NULL,
startStationId INTEGER NOT NULL,
endStationId INTEGER NOT NULL,
FOREIGN KEY (startStationId) REFERENCES RailwayStations(stationID),
FOREIGN KEY (endStationID) REFERENCES RailwayStations(stationID),
FOREIGN KEY (bedID) REFERENCES Bed(bedID),
FOREIGN KEY (orderID) REFERENCES CustomerOrders(orderID),
FOREIGN KEY (occurrenceID) REFERENCES TrainOccurrence(occurrenceID),
CONSTRAINT uniqueTicket UNIQUE (occurrenceID, bedID)
);

```

Eksempler

```

INSERT INTO RailwayStations(stationID, stationName,altitude)
VALUES(1,"Trondheim", 5.1),
(2,"Steinkjer", 3.6),
(3, "Mosjøen", 6.8),
(4,"Mo i Rana", 3.5),
(5,"Fauske", 34.0),
(6,"Bodø", 4.1);

```

```

INSERT INTO TrackSection(sectionID, sectionName,drivingEnergy,startStationId,
endStationID )
VALUES (1,"Nordland", "Diesel",1,6);

```

```

INSERT INTO SubSections (subSectionID, subSectionLength_km, trackType, trackSectionID,
startStationId, endStationID)
VALUES (1, 120, "double", 1, 1, 2),
(2, 280, "single",1, 2, 3),
(3, 90, "single", 1,3, 4),
(4, 170, "single",1, 4, 5),
(5, 60, "single", 1,5, 6);

```

```
INSERT INTO Operator (operatorID, operatorName)
VALUES (1, 'SJ');
```

```
INSERT INTO CarType (carTypeID, operatorID, typeName)
VALUES (1, 1, 'Chair Car'), (2, 1, 'Sleeping Car');
```

```
INSERT INTO ChairCar (chairCarId, carTypeID, OperatorID, numberOfSeatRows,
numberOfSeatsPerRow)
VALUES (1, 1, 1, 3, 4);
```

```
INSERT INTO SleepingCar (sleepingCarID, carTypeID, operatorID, numberOfCompartments)
VALUES (1, 2, 1, 4);
```

```
INSERT INTO SleepingCompartment (compartmentID, sleepingCarID, compartmentNumber,
numberOfBeds, OperatorID)
VALUES (1, 1, 1, 2, 1), (2, 1, 2, 2, 1), (3, 1, 3, 2, 1), (4, 1, 4, 2, 1);
```

```
INSERT INTO Bed (bedID, compartmentID, bedNumber, bedType)
VALUES (1, 1, 1, 'Lower'), (2, 1, 2, 'Upper'), (3, 2, 1, 'Lower'), (4, 2, 2, 'Upper'), (5, 3, 1,
'Lower'), (6, 3, 2, 'Upper'), (7, 4, 1, 'Lower'), (8, 4, 2, 'Upper');
```

```
INSERT INTO Seat (seatID, seatNr, rowNumber, chairCarID, operatorID)
VALUES (1, 1, 1, 1, 1), (2, 2, 1, 1, 1), (3, 3, 1, 1, 1), (4, 4, 1, 1, 1),
(5, 5, 2, 1, 1), (6, 6, 2, 1, 1), (7, 7, 2, 1, 1), (8, 8, 2, 1, 1),
(9, 9, 3, 1, 1), (10, 10, 3, 1, 1), (11, 11, 3, 1, 1), (12, 12, 3, 1, 1);
```